The Motor Industry Software Reliability Association

MISRA c/o Electrical Group, MIRA, Watling Street, Nuneaton, Warwickshire, CV10 0TU, UK.
Telephone: (024) 7635 5290. Fax: (024) 7635 5070. E-mail: misra@mira.co.uk Internet: http://www.misra.org.uk

Report 4

Software in Control Systems

February 1995

PDF version 1.0, January 2001

Acknowledgements

The following were contributors to this report:

Keith Longmore, Lotus Engineering
David Newman, Ford Motor Company Ltd
Mike Radford, Lucas Electronics
David Ward, MIRA

Summary

This report examines the role of software in the design of control systems. It is divided into three major parts:

- theoretical considerations
- design considerations
- practical considerations.

It is recommended that textbooks on control are read for detailed discussion of the theory. A recommended reference list is appended.

Theory is split into linear and nonlinear control. The former is the simpler, and most readily understood.

Differential equations in the time domain, and Laplace and *z* transforms in the frequency domain, play a major part in control theory. Stability and damping are critical to the performance of a control system. Various methods exist for calculation of stability.

In the context of theoretical summarization, no particular recommendation is made regarding methods to be used for calculating stability.

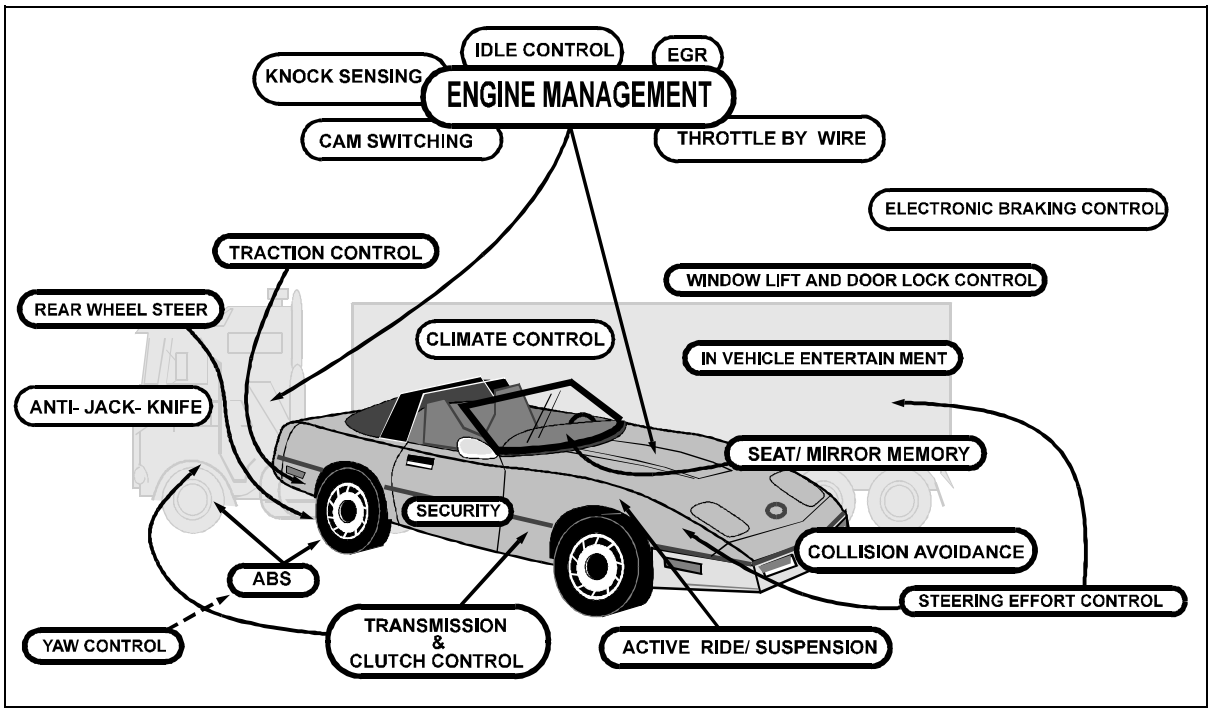


Figure I The scope for software-based control systems in vehicles

Recommendations

Theoretical considerations

Control theory

· In order to create a comprehensive requirements specification for a control system, it is essential to have a thorough understanding of the complexities and mathematics of the relevant control theory.

Accepted control techniques

· Several control techniques may be required in a complex control application. Selection should be on the basis of the following characteristics:

- predictability
- defined target hardware resources
- deviations between implementation and theory
- robustness (especially of input data)
- testability
- simplicity.

Sampling and aliasing

· Sampling is an inherent part of digital control techniques. Attention should be paid to the resulting issue of aliasing; in particular, the sampling rate should be high enough to minimize aliasing.

· If the sampling rate is software controlled, software timing should not be based on loop execution time alone.

· Both resolution and system response times should meet the system requirements.

· Consider the effect of quantization. If dithering is used to reduce the quantization errors, consider its effect on system response, accuracy and linearity.

· Take adequate design precautions to reduce the effects of noise signals (e.g. anti-aliasing filters).

· Evaluate the effects on closed-loop control of:

- input filters
- output filters
- controlled object response.

Open-loop control

· Open-loop control has widespread application in vehicle control systems during:

  - a system development phase, for quantifying parameters
  - initialization, to aid in checking system component health
  - a default or "limp home" condition
  - transient system conditions, where the speed of response cannot be provided by closed-loop action.

· Care should be taken in mapping parameters for open-loop control. In particular, the effects of sensor errors may be more pronounced than in a closed-loop system. Any development aids used in the calibration of open-loop systems should be designed for ease of use.

· Calibration is a skilled activity. Experience is essential for the achievement of optimum performance.

Design considerations

Requirements specification

· Give special consideration to the required attributes of a control system at the requirements specification stage. Performance attributes should be accurately quantified. Assumptions, uncertainties and unknowns should be identified and documented in the requirements specification.

Stability

· The selected method or methods for the calculation of control system performance, and especially stability, should be defined in the requirements specification. It is particularly important to be consistent where parts of a system are created by different teams.

· Parameters critical to system stability should be identified in the requirements specification.

Transient conditions

· The definition of transient conditions requires particular care.

Formal specification

· Formal mathematical methods, especially for specification, may appear appropriate for this subject. However, control theory already has its own mathematical basis,

therefore the benefits of formal mathematical methods may be outweighed by added complexity when attempting to apply them in this area. Many control systems incorporate concurrency, which is not yet widely supported by current industrial strength formal mathematical methods.

Graceful degradation

· A control system should be specified such that it degrades in a graceful manner in accordance with its integrity and availability requirements.

Fault recovery

· Fault management routines should be designed and demonstrated to respond within timing requirements for critical routines.

· Fault migration between functions should be minimized by partitioning and recovery block schemes. Recovery from a fault management routine should be demonstrated, not assumed.

· In multiprocessor systems, fault recovery action should be synchronized and prioritized, especially where communications failure is detected.

· It is essential that recovery action is not inhibited by interrupts.

Diagnostics

· Failure management should offer alternative sensor information or mechanical back up wherever possible, and all default definitions should be supported by a well-reasoned diagnostic strategy and objectives.

Software security

· There should be protection against unauthorized access to software. Various methods are available.

· An alternative is to provide for the detection of tampering, for example as in US CARB OBD II legislation. As a minimum, ensure that the evidence of tampering is clearly apparent.

Accuracy

· It is important to consider carefully the implications of:

- dynamic range
- linearity
- conversion time

- response time
- noise
- damping
- effects of arithmetic systems used (e.g. fixed, floating point)
- effect of accumulation of errors and rounding.

· In order to perform accurate control, it is important to acquire accurate knowledge regarding the state of the system being controlled. This can be difficult because of the invisibility of some data, lag in the system, or poor understanding of the control actions needed. This can result in poor accuracy, compensated for by the manual adjustment of parameters during development. However, there are techniques that may be employed to improve control accuracy:

- inferred variables
- statistical estimates
- artificial variables
- observers
- Kalman filters
- feedforward control
- predictive control.

· It is important to carefully consider requirements for:
- dynamic range
- linearity
- conversion time
- response time
- noise
- damping
- effects of using available arithmetic systems (e.g. fixed, floating point)
- effect of accumulation of errors and rounding.

Optimization

· Optimization of control is the process of developing a control system to be as efficient as possible within defined constraints. In relatively linear systems, good optimization may be achievable by the application of theory or modelling. In the case of significantly nonlinear systems, however, optimization is often not straightforward, and it is important to consider the following factors:

(a) In significantly nonlinear systems it is possible for the input fundamental frequency not to be present in the output.

(b) In nonlinear systems there may be local nonlinearities, which are impossible to analyse and quantify; thus optimization may have to be done by empirical methods.

(c) The most significant problem of nonlinear control systems is that the dynamic behaviour becomes a function of amplitude.

Adaptive control

· Adaptive control is effectively continuous on-line optimization of parameters in a closed-loop control system, but its use should be assessed in relation to the possible failure modes and range of authority.

· As few variables as possible should be used in the adaptation algorithm, in order to restrict the number of degrees of freedom.

· Even where adaptation is used, it may still be necessary to use manually derived look-up tables in the adaptation algorithm to compensate for significant nonlinearities.

Neural networks

· Neural networks are a special case of adaptive control in which the network, by a process of feedforward prediction, "learns" about the device that it is to control using a "learning" algorithm and real recorded data.

· Neural networks are effective at handling nonlinearities, and especially systems that have poor visibility for some, possibly critical, parameters.

· Neural networks may be very difficult indeed to validate. It may also be difficult to demonstrate the stability of control systems using neural networks under all operating conditions.

Diagnostics and fallbacks

· Some possibilities for handling predicted component failure are:

- a policy of service replacement
- standard component performance test, comparing with original data
- multiple channel, redundant systems or components
- inference from other sensors
- inference from start up data derived from setting components to known states
- monitoring trends.

Safe states, redundancy, and diversity

· In contrast to open-loop systems, and in common with all digital systems, closed-loop control systems work very reliably, but tend to fail dramatically. Therefore, it is important to emphasize the design and validation of the failure management mechanisms.

- There is a high risk of "fault masking" in microprocessor based closed-loop control systems. This occurs when the failure mode management is so effective that the driver fails to recognize the presence of a fault. The use of condition monitoring and appropriate warnings are recommended.

- It is recommended that a combination of failure management techniques are used for handling failures in closed-loop systems, including switching to open-loop operation.

Practical considerations

Modelling

- The use of a modelling tool is recommended to aid the design of control systems. There are a significant number of commercial modelling packages available, as well as custom designed and maintained packages.

- Give careful consideration to the choice between commercially available and custom modelling packages. Commercial packages offer great flexibility and ease of use; custom packages may be much more powerful for the given application for which they are designed.

- Modelling packages may be used as a validation tool provided adequate confidence in the package itself can be justified.

Simulation and emulation

- Simulation and emulation can be beneficial to the development process.

- In order to have sufficient confidence in the results of simulation and emulation, software quality management should be applied to them.

- If emulation techniques are to be used as part of the prototype stage, manufacturers should satisfy themselves of the safety of the emulation in advance. It should allow inspection, and not modification, of control algorithms.

Contents

Page

1. Introduction

1.1 Report format

This report is divided into three major parts:

- theoretical considerations
- design considerations
- practical considerations

Part I — Theoretical considerations

2. Introduction

The purpose of this Part is to define control theory and its place in the development of safe, reliable software. Particular attention is given to the requirements specification phase of the software development life cycle, so that the control of a function may be adequately designed and the correct parameters supplied.

An introduction is given to some of the concepts and techniques which may be encountered. As this is necessarily brief, readers are invited to consult the list of references given for further details.

It is assumed that those involved are practising control engineers with evidence of appropriate education.

3. What is control theory?

The subject of "control" is very broad, but may be roughly divided into "control theory" and "control techniques". The latter are discussed in Section 6.

Control theory is the mathematical description of the behaviour of systems and the means by which the future behaviour of those systems may be altered. The term system is used in a very general sense to describe a collection of interacting elements which are connected by links across which information may travel.

Control theory is generally applied to dynamic systems, that is, those which have a behaviour that varies with time. Such systems are not always mechanical or physical; electrical and thermal systems are two examples. The most readily studied systems are linear. This means that if a system has a known response to a given stimulus, the response to a scaled version of that stimulus is similarly scaled **with proportionality preserved**. However, many real-life situations are highly non-linear and particular strategies must be used to handle them.

Control theory is typically applied to feedback systems, where a proportion of output or outputs is returned as inputs to the system. Feedback control is also known as closed-loop control. In the real world, however, situations arise where the use must be considered of both open-loop and closed-loop control within one system .

It is necessary to analyse the system to determine its stability (or otherwise) and the conditions, if any, under which the system is stable. Although much of the theory is initially applicable to continuous systems, many real life controllers are discrete, that is, the time-varying functions are defined at specific intervals only rather than at all times. This section is primarily concerned with continuous functions, although references are made to some discrete theories. The subject is more completely covered in later sections.

4. Linear control theory

Linear systems are the easiest to study. Mathematically, a function *f* is linear if

$$f(u_1 + u_2) = f(u_1) + f(u_2) \quad \textbf{(1)}$$

for any u_1 and u_2 in the domain of *f* and if

$$f(\alpha u) = \alpha f(u) \quad \textbf{(2)}$$

for any *u* in the domain of *f* and any real number α.

Linear systems may be analysed in either the time domain or the frequency domain.

4.1 Time domain

4.1.1 Differential equations

The dynamics of a system may be expressed as one or more differential equations, that is, equations which contain a derivative term or terms. The general form of a differential equation is

$$a_n \frac{\mathrm{d}^n y}{\mathrm{d}t^n} + a_{n-1} \frac{\mathrm{d}^{n-1} y}{\mathrm{d}t^{n-1}} + \cdots + a_0 y = b_r \frac{\mathrm{d}^r u}{\mathrm{d}t^r} + b_{r-1} \frac{\mathrm{d}^{r-1} u}{\mathrm{d}t^{r-1}} + \cdots + b_0 r + c \quad \textbf{(3)}$$

where *y* is the output of the system and *u* the input. The differential equation may be directly solved in the time domain for certain cases, notably a second order equation of the form

$$a_2 \frac{\mathrm{d}^2 y}{\mathrm{d}t^2} + a_1 \frac{\mathrm{d}y}{\mathrm{d}t} + a_0 y = f(t) \quad \textbf{(4)}$$

First, the complementary function is found by setting $f(t) = 0$. Writing down the auxiliary equation gives

$$a_2 \lambda^2 + a_1 \lambda + a_0 = 0 \quad \textbf{(5)}$$

which has roots λ_1 and λ_2. If the roots are different and real, the solution is of the form

$$x = k_1 e^{\lambda_1 t} + k_2 e^{\lambda_2 t} \quad \textbf{(6)}$$

and decays exponentially (overdamped). If the roots are coincident, the solution is of the form

$$x = (k_1 + k_2 t) e^{\lambda t} \quad \textbf{(7)}$$

and is termed critically damped. If the roots are complex, $\lambda = \alpha \pm j\beta$, the solution is of the form

$$x = (k_1 \cos\beta t + k_2 \sin\beta t)e^{\alpha t} \quad \textbf{(8)}$$

and is underdamped (oscillatory). The particular integral is then determined by assuming the general form of $f(t)$.

Examination of the solutions of differential equations can give information about the stability of the system, but is limited by the solution techniques available. For more general cases the transfer function has to be determined (q.v.)

4.1.2 Difference equations

For a discrete system the differential equation is replaced by a difference equation of the general form

$$a_n y(i+n) + a_{n-1} y(i+n-1) + \cdots + a_0 y(i) = b_r u(i+r) + b_{r-1} u(i+r-1) + \cdots + b_0 r(i) + c \quad \textbf{(9)}$$

Solution of equations of this form follows similar methods to those used for differential equations.

4.2 Frequency domain

4.2.1 Laplace transforms

The Laplace transform forms the basis of many important techniques. The Laplace transform $\mathcal{L}$ of a function $f(t)$ is signified by the notation

$$F(s) = \mathcal{L}\{f(t)\} \quad \textbf{(10)}$$

The Laplace transform is by definition

$$F(s) = \int_0^\infty \exp(-st) f(t)\,dt \quad \textbf{(11)}$$

Note that s is a complex variable.

4.2.1.1 z transforms

The z transform is the discrete equivalent of the Laplace transform, and is in fact derived from it [1]. The z transform $F(z)$ of a sampled version $f(n)$ of $f(t)$ is defined as

$$Z\{f(n)\} = F(z) = \sum_{n=0}^{\infty} f(n)z^{-n} \quad \textbf{(12)}$$

4.2.1.2 Transfer function

The Laplace transform leads to the definition of the transfer function of a system. The transfer function $G(s)$ of a system with input $u(t)$ and output $y(t)$ is **defined** to be

$$G(s) = \frac{Y(s)}{U(s)} \quad \textbf{(13)}$$

provided both transforms exist.

Consider the general differential equation (3). The Laplace transform, neglecting terms on both sides due to initial conditions, is

$$(a_n s^n + a_{n-1}s^{n-1} + \cdots + a_0)Y(s) = (b_r s^r + b_{r-1}s^{r-1} + \cdots + b_0)U(s) \quad \textbf{(14)}$$

The transfer function is simply given by

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_r s^r + b_{r-1}s^{r-1} + \cdots + b_0}{a_n s^n + a_{n-1}s^{n-1} + \cdots + a_0} \quad \textbf{(15)}$$

Complex number analysis techniques when applied to the transfer function can yield much useful information. Pole-zero analysis is an example. A zero of $G(s)$ is a value of the complex variable s for which $G(s) = 0$, and a pole of $G(s)$ a value s_1 for which $G(s) \rightarrow \infty$ as $s \rightarrow s_1$. Clearly for a transfer function in the form above the poles are the roots of the equation $U(s) = 0$ and the zeros the roots of the equation $Y(s) = 0$. The poles and zeros may be plotted on an Argand diagram and the Nyquist criterion used to assess stability. The standard texts [2, 3] describe the Nyquist criterion and its method of application.

For a feedback control system it is common for the overall transfer function to be composed of the product of the individual transfer functions for the controller and the process.

5. Non-linear control theory

In practice many systems are non-linear and particular methods must be used for analysis.

5.1 Describing functions

The describing function method is specifically applicable to non-linear systems which can be decomposed into a non-linear function *f* followed by a linear transfer function *G*(*s*). Assume a sinusoidal signal $u = a\sin\omega t$ is applied to the function *f*. The output can be expressed as a Fourier expansion:

$$f(a\sin\omega t) = a_0 + \sum_{k=1}^{\infty}(a_k\cos k\omega t + b_k\sin k\omega t) \quad \textbf{(16)}$$

Thus the output of the non-linear element is composed of the following components:

- a "DC" or mean level a_0
- a fundamental $a_1\sin\omega t + b_1\cos\omega t$
- harmonics at frequencies 2ω, 3ω, etc.

If all harmonics except the fundamental are neglected, the non-linear function *f* can be expressed by an equivalent gain, which is denoted *N*(*a*). It is defined by

$$\begin{aligned} N(a) &= \frac{a_1\sin\omega t + b_1\cos\omega t}{a\sin\omega t} \\ &= \frac{a_1}{a} + j\frac{b_1}{a} \end{aligned} \quad \textbf{(17)}$$

N(*a*) is thus a complex number and may be displayed as a locus on an Argand diagram. The steady-state sinusoidal behaviour of the system is then given by $G(j\omega)N(a) = -1$, or

$$G(j\omega) = \frac{-1}{N(a)} \quad \textbf{(18)}$$

The loci of $G(j\omega)$ and $-1/N(a)$ may be plotted on the same diagram and the Nyquist criterion applied to assess the stability of the loop.

5.2 Phase plane portrait

For a **linear** second-order system with zero input, the process may be expressed as

$$\begin{aligned} \dot{x}_1 &= a_{11}x_1 + a_{12}x_2 \\ \dot{x}_2 &= a_{21}x_1 + a_{22}x_2 \end{aligned} \quad \textbf{(19)}$$

with the x_i denoting state variables and the a_{ij} are elements of a matrix ***A*** defining the relationship between the state variables, namely

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} \quad \textbf{(20)}$$

Solutions of this equation, with $x(0) = x_0$ may be plotted in the state plane with time as a parameter which varies along them and are known as trajectories. A combination of the two is known as a state plane portrait.

Examination of the trajectories gives information on the stability of otherwise of the system. If the system is stable, then as time increases the trajectories approach or reach the origin of state space. Conversely, if the system is unstable the trajectories start from the origin and move away from it.

The matrix $\boldsymbol{A}$ has eigenvalues λ_i which are solutions of the equation $|\lambda \boldsymbol{I} - \boldsymbol{A}| = 0$. These eigenvalues are associated with eigenvectors $\boldsymbol{e}_i$ which are solutions of the equations $\boldsymbol{A}\boldsymbol{e}_i = \lambda_i \boldsymbol{e}_i$. In physical terms, the eigenvectors correspond to the normal modes of oscillation of a system. The eigenvalues and eigenvectors give important information about the solutions of the system matrix and characterize the global behaviour.

For non-linear systems, the state plane consists of a plot of trajectories in the x_1-x_2 plane of a second order process of the form

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2) \\ \dot{x}_2 &= f_2(x_1, x_2) \end{aligned} \quad \textbf{(21)}$$

In the special case where x_2 is the derivative of x_1 the name phase plane is used, which gives rise to the phase plane portrait. The system will have a number of critical points denoted c_1, c_2, ... c_n which are solutions of the equations $f_1(x_1, x_2) = f_2(x_1, x_2) = 0$.

As the process is non-linear, linearization must be applied to obtain the matrix $\boldsymbol{A}$ (see below). A typical element will have the form

$$a_{ij} = \frac{\partial f_i}{\partial x_j} \quad \textbf{(22)}$$

The derivative may be viewed as a tangent plane to the state surface at the point of interest, and the matrix as a linear transformation that approximates f close to x. The matrix is called a Jacobian matrix. There will generally be a different matrix for each critical point $\boldsymbol{A}_{C_1}$, $\boldsymbol{A}_{C_2}$, ... $\boldsymbol{A}_{C_n}$ which are found by solving the general expression for the $\boldsymbol{A}$ matrix at each critical point.

The actual system behaviour in a small region surrounding each critical point will be governed by the eigenvalues and eigenvectors of the appropriate $\boldsymbol{A}$ matrix. It is thus straightforward to determine the behaviour of the non-linear system in the regions close to critical points, and often by continuing the solutions so that they join in a sensible fashion the complete behaviour may be obtained.

5.3 Linearization

Linearization is the replacement of a non-linear function by a linear approximation, allowing analysis of a non-linear problem by linear techniques. The use of the phase plane portrait described above is an example of such analysis. Care must be taken when interpreting the results of such an analysis to ensure that unacceptably large errors are not caused by the linearizing approximation.

The basic linearization technique is the familiar Taylor expansion of an analytic function. Taylor's theorem (for an analytic function of one variable) is

$$f(a+h) = f(a) + h\left.\frac{\mathrm{d}f}{\mathrm{d}h}\right|_a + h^2\left.\frac{\mathrm{d}^2 f}{\mathrm{d}h^2}\right|_a + \ldots \qquad \textbf{(23)}$$

Here h is considered to be the perturbation about a fixed point a at which the linearization is performed. To linearize, the higher-order terms in the expansion are neglected, so that the function f about the point a is replaced with the approximation

$$f(a+h) \approx f(a) + h\left.\frac{\mathrm{d}f}{\mathrm{d}h}\right|_a \qquad \textbf{(24)}$$

The accuracy of this approximation depends on

- the magnitude of h
- the magnitude of the higher derivatives of f at $x = a$
- the behaviour of $\mathrm{d}f/\mathrm{d}h$ at $x = a$.

If the function f operates on a vector Taylor's expansion still applies: now $f(\boldsymbol{a}+\boldsymbol{h}) \approx f(\boldsymbol{a}) + \boldsymbol{J}\boldsymbol{h}$ where the matrix $\boldsymbol{J}$ is defined as

$$j_{ij} = \left.\frac{\partial f_i}{\partial h_j}\right|_a \qquad \textbf{(25)}$$

Linearization is not limited to a first-order Taylor expansion: other methods include

- linearization about an analytically specified trajectory
- describing functions (q.v.)

For further details the standard texts should be consulted.

5.4 Lyapunov's second theorem

Lyapunov's second theorem (or direct theorem) is a means of assessing stability. The concept of a Lyapunov function is introduced. A function V is called a Lyapunov function on a region U of the n-dimensional state space X (which must contain the origin) of a system if

- V is positive definite and strictly increasing on U
- $\dot{V}$ is negative semi-definite on U

See a standard mathematics text for definitions of these properties. Lyapunov's second stability function states: "if on a region U, containing the origin of X, there exists a Lyapunov function for which $\dot{V} < 0$ on U, then the origin is a stable critical point and all solutions originating in the region U approach the origin asymptotically" [4].

It is possible to bring any critical point to the origin of phase space and so apply the Lyapunov theorem by a change of axes.

Conversely, if

- V is positive definite on U
- $\dot{V}$ is positive definite on U

then the origin is an unstable critical point. This is known as the Cataev instability theorem.

A major disadvantage of Lyapunov's second theorem is the need for a Lyapunov function to exist to ensure stability. The standard texts (for example [4]) give guidelines on choosing a suitable function.

Lyapunov's direct method is so-called due to the existence of the Lyapunov first (or indirect) method. In the first method, the Lyapunov function is applied to the linear part of the system only; in the second method, directly to the non-linear equations themselves.

5.5 Envelope methods

Envelope methods are means of assessing the stability of a feedback loop consisting of a non-linearity followed by a stable transfer function. The non-linearity is enclosed in a linear envelope and the latter is used in the subsequent analysis. The results of the analysis lead to sufficient but not necessary stability conditions.

The Popov stability criterion is a graphical criterion for assessing the stability of the loop just described and is akin to the Nyquist criterion. A modified transfer function $G^*(j\omega)$ is defined by

$$G^*(j\omega) = \mathbf{Re}[G(j\omega)] + j\omega \mathrm{Im}[G(j\omega)] \qquad \textbf{(26)}$$

The locus of this modified transfer function (known as the Popov locus) is plotted on an Argand diagram, together with a straight line representing the linear envelope. The straight line intersects the real axis at a point -1/*k* which defines the envelope. The system is stable if the straight line does not intersect the Popov locus.

The circle method is a generalization of Popov's method which allows

- $G(s)$ to be open-loop unstable
- the non-linearity to vary with time

The non-linearity is enclosed within an envelope defined by two straight lines intersecting at the origin. The system is asymptotically stable if the Nyquist plot $G(j\omega)$ lies outside a circle in the complex plane defined by a standard relation given in the texts.

6. Accepted control techniques

6.1 Scope

In view of the complexity and depth of control theory and techniques it is not possible to recommend one particular control method over another for a particular problem area. This section aims to point out potential pitfalls to consider when implementing control laws in predominantly software based electronic control systems.

6.2 Discussion

The control method to be implemented must be well understood (both in academic and practical terms) by the project team.

From an understanding of the control required a suitable technique must be selected. The techniques might range from simple PID controllers to multivariable systems with "observers". See Figure 1.

The particular technique selected must have the following attributes:

- It must allow the controller design to have sufficient capability to control the system to the requirements of the theoretical studies. (This may be assessed by simulation of the controller. Knowledge of similar, working systems is also useful.)

- The implementation of the controller algorithm (on the target processor) will introduce distortions (numeric representation errors, round off errors, time delays, etc.) Where possible the effect of these distortions should be assessed

System	Open-loop	P	I	D	Adaptive	Bang bang	Lin	Nonlin
Fuelling	x	x	x	x	x	x	x	x
Knock control	(x)					x		
EGR	(x)	x	x	x				x
Idle control		x	x	(x)	x			x
Turbo control		x	x	x				x
"Throttle by wire"		x	x		(x)		x	
ABS		x	x			x		x
Active suspension	x	x	x	x	(x)		x	x
Transmission control					(x)	x		x
Clutch control		x	(x)		(x)			x
Ride height control		x	x		x			
Air conditioning		x	x	x			x	x
Traction control		x		(x)				x
CVT		x					x	
PAS control		x					x	
Cruise control		x	x		(x)			

(x) indicates conditional or possible current use in a control algorithm.

Figure 1 Typical examples of automotive systems versus control algorithm type

during the control technique selection process. This might be achieved by modification of the idealized control technique simulation to model such implementation artifacts. This simulation process may point to the need, for example, for floating point hardware to achieve the required accuracy and performance.

The control technique (and subsequent implementation) must have the following characteristics:

· Predictability
- The algorithm must have a defined upper limit on execution time. Preferably

the execution time (i.e. number of operations) should not be related to data values.
- The algorithm employed must have a defined upper limit on temporary storage requirements. This limit should include stack requirements of the low level implementation as well as temporary variables dictated by the control technique (i.e. intermediate variables).

· Implementation (i.e. approximations in implementation vs. theory)
- Know what the implementation has done to the "pure" control technique in terms of limited numeric range, limited accuracy and resolution, algorithm modifications for speed/memory savings, etc. If at all possible these constraints should be used as inputs to the modelling process so that their impacts can be estimated and accommodated within the proposed controller design.

· Robustness of input data
- The method/algorithm selected must be robust (in time and space requirements) for all possible input data values. Note that this does not just include "normal operation" values but inputs produced as the result of errors in earlier processes (such as reading of sensors). This is particularly important in the case of safety-related systems. A safety-critical control loop component should check input data against its operational limits and have a defined error flagging and reversionary scheme for inputs outside of this range. Note that maps provide input data for many systems and these must be generated, controlled, checked and use in a rigorous fashion (see §8.3 "Mapping of open-loop".)

· Testability
- The technique chosen must be adequately verifiable (both on the bench and on the vehicle) by available test tools. If the verification of a design requires access to large numbers of variables in real-time this may mandate against its selection.

· Simplicity
- If two techniques offer similar performance then the simplest one should be chosen. Selecting complex techniques will increase the chance of errors going undetected during the development lifecycle.

7. Sampling and aliasing

7.1 Introduction

The purpose of this section is to examine the implications of digital control of feedback systems, with particular reference to sampling and aliasing.

An introduction is given to some of the concepts and techniques which may be encountered. As this is necessarily brief, readers are invited to consult the list of references given for further details.

It is assumed that those involved are practising control engineers with evidence of appropriate education [5, §7].

7.2 Discrete systems

A continuous system is a system that can change its conditions progressively, whereas a discrete system can change its state only at set points. Where a system can only change its state at discrete times, it is known as a discrete time system.

Many real-life systems are continuous, and continuous feedback control may be accomplished by constructing the appropriate analogue hardware. However, the flexibility of software means that many systems are now under computer control. Traditionally, the term "computer" has covered both analogue and digital computers, but the former are now so rare that "computer" is almost universally synonymous with "digital computer" (and will be adopted for the remainder of the present section).

Computers offer a great deal of flexibility in the control of systems, but input signals must be converted from analogue to digital form and output signals back from digital to analogue. Consequently there are **two** aspects to discrete system control:

- discrete control algorithms
- digitization of continuous signals.

The difficulties associated with digital control occur in both of these areas. There are two routes to obtaining discrete control algorithms:

- recast continuous control algorithms
- design from the outset as discrete algorithms.

7.2.1 Discrete control algorithms

Discrete time algorithms operate on a sequence of error signals to produce a sequence of command signals. An algorithm may be specifically designed as discrete from the outset, but it is more likely that the state equations of a continuous system will need to be recast in discrete form. A number of the techniques are discussed briefly below.

7.2.1.1 Differential to difference equations

The general form of a differential equation is given in equation (3). It may be directly transformed by a suitable method into an equivalent difference equation of the general form given in equation (9). Solution of equations of this form follows similar methods to those

used for differential equations. Note that the difference equation is a discrete form of the differential equation for a chosen time interval *T*. The difference equation may be used to generate an approximate numerical solution for the original differential equation, assuming that the initial conditions for the differential equation are known and can be related to the starting values required for the difference equation. In practice, this may not always be the case.

7.2.1.2 s domain to z domain

The Laplace transform is often used to study the behaviour of continuous systems and is defined in equations (10), (11). The *z* transform is the discrete equivalent of the Laplace transform, and is in fact derived from it [1]. It is possible to transform a continuous system to a discrete system by substituting *s* with its equivalent function in *z*. However, as $z = \exp(sT)$, the substitution required would be $s = \ln z/T$ which would lead to an undesirable polynomial in ln *z*.

Note that the *s* domain is infinite, but maps to the *z* domain in a periodic fashion as may be seen from an examination of the complex function $\exp(sT)$. This phenomenon may be closely identified with frequency folding (see below).

7.2.1.3 Example techniques

A system may be treated in a completely digital fashion (excluding the plant), or some form of mapping may be used. Examples of techniques are:

- finite difference approximations: derivatives dy/dt are replaced with finite difference approximations, such as $(y_{k+1} - y_k)/T$. This is a low frequency technique
- state space techniques (effectively a subset of the previous), where linear state space equations may be used to obtain a difference equation
- mapping the poles of a continuous transfer function to the correct equivalent points in the *z* plane. This is also a low frequency technique
- using the relation

$$G(z) = Z\{\mathcal{L}^{-1}[G(s)]\} \qquad \textbf{(27)}$$

  for impulse response mapping
- multivariable analysis: convert $G(s)$ into multivariable form $\{A, B, C\}$ and use $\Phi(T)$ and $\Psi(T)$ as discrete operators [4]
- bilinear transforms (Tustin's approximation): given a compensation $D(s)$ in a continuous control system, the substitution

$$s = \frac{2(z-1)}{T(z+1)} \qquad \textbf{(28)}$$

  in any $D(s)$ yields a $D(z)$ based on the trapezoidal integration formula.

However, this technique introduces a frequency distortion which must be corrected. The correction is usually achieved by applying a "prewarp" so that the effects of the distortion are removed [7, p. 52]

- a standard numerical algorithm for the time solution of differential equations, provided the computational overheads are not unacceptably large. For example Runge-Kutta methods can be used to solve differential equations, but the associated computational overhead would preclude its use in a real-time system.

For further details standard texts should be consulted.

7.2.2 Digitization of signals

A signal is digitized by sampling at (normally fixed) intervals *T*. This process will usually be performed by an analogue to digital (A/D) converter. The resultant signal is both discrete, namely discontinuous in time, and quantized. The signal is quantized because the output of an A/D converter must be represented by a digital word composed of a finite number of binary digits (bits). Both the sampling interval *T* and the quantization *q* must be taken into account when designing the system. Aliasing is an example of the problems that can be associated with the choice of the sampling interval.

7.2.2.1 Aliasing and folding

As well as issues of accuracy and scaling (see §14) the problem of aliasing has to be considered. The analogue domain is cyclically mapped in terms of frequency to the digital domain, which is known as "folding". Aliasing occurs when too low a sampling frequency causes the resulting data to contain additional low frequency components that were not present in the original signal, due to this folding.

An example of aliasing is shown in Figure 2. An oscillatory signal with a frequency of 60 Hz is being sampled at intervals corresponding to 50 Hz. The Figure shows that the resulting sampled signal has a frequency of 10 Hz and also shows the mechanism by which the aliasing occurs.

An important theorem is Nyquist's sampling theorem. To uniquely represent the analogue input signal without aliasing, the sampling rate must be at least twice the maximum frequency component of the signal being sampled. In the example, the sampling frequency has to be at least 120 Hz. However, a sampling rate of exactly twice the maximum frequency would give no gain or phase information, and in practice the sampling rate has to be higher (see below).

Aliasing can have a substantial effect on a digital control system. Noise components, which normally have a frequency higher than the bandwidth of the control system, can be aliased down and cause the system to respond, with the noise appearing on the output of the system as an in-band frequency. The bandwidth is assumed to mean the frequency at which the output of a system is reduced by 3 dB compared to a reference point (often DC).

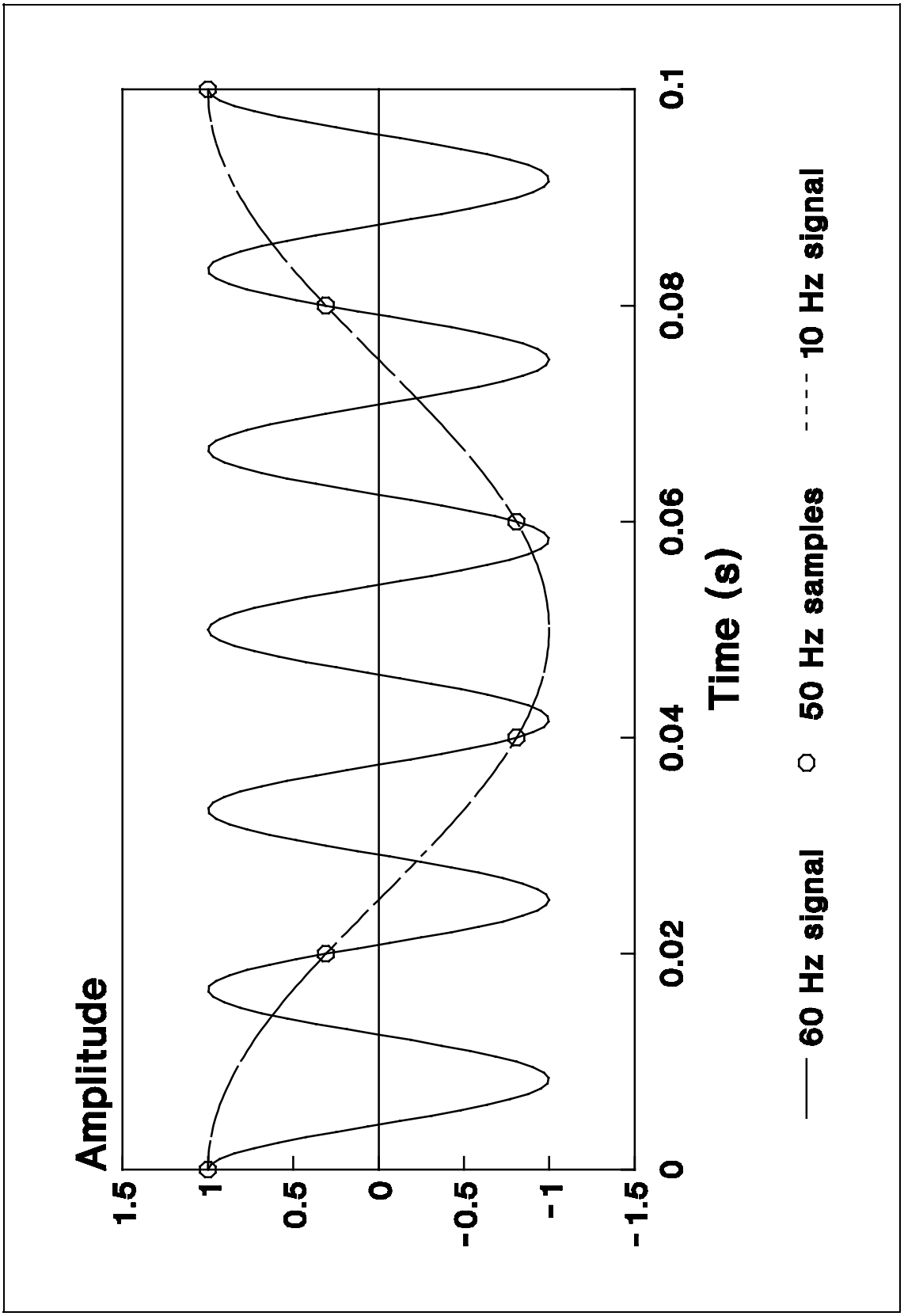


Figure 2 Example of aliasing

A common solution to noise problems is to place an analogue filter between the sensor and the A/D converter. In most cases a simple first-order low-pass filter is found to be adequate. Filtration does not eliminate aliasing, but a careful choice of the cutoff frequency and the sampling rate can reduce aliasing substantially. However, filters may have undesirable effects on closed-loop behaviour and must be included in the analysis.

The effect of "jitter" on sampling should also be considered. Jitter occurs if the sampling interval varies slightly from its nominal value. The classic example is with the sampling of a triangle wave, where a variation of even 1% in the sample period can lead to significant effects on the sampled waveform, especially close to the apex. Jitter is unlikely to be experienced if the sampling period is derived from a stable hardware source, such as a crystal-backed clock, but can occur if the sample rate is set in software. If software control of sampling rates is desired, software timing should not be used and the sample rate should be set via a divided clock frequency.

7.2.2.2 Other sampling methods

It is possible to use a number of variations on uniform sampling intervals, which require an extension of the analytical methods. Full details are beyond the scope of this section, and readers are invited to consult appropriate references. The techniques which may be encountered include:

- multirate sampling, where a system requires two or more different sample rates to be used for different signals. The system may be designed using successive loop closure [3], but advances in computer performance mean that the highest sampling rate can now usually be applied across the whole system
- skip rate sampling
- cyclic rate sampling, where two periods T_1, T_2 between samples are used alternatively T_1, T_2, T_1, T_2, ...
- asynchronous sampling, for example where a second sampling operation occurs in the feedback loop and the times of the second samples do not coincide with the first
- non-uniform sampling (variable step size), for instance in an engine management system where the sampling rate may be related to the engine speed. It is common for sampling to be synchronous to the process being controlled
- software control of sampling intervals: the effects of jitter (see above) should be considered.

Generally the texts recommend the use of the state variable technique or polynomial analysis for dealing with the majority of the above sampling methods.

7.2.2.3 Choice of sampling rate

The choice of the best sampling rate is inevitably a compromise between the cost of the

system and its performance. The cost will increase with higher sampling rates, but the performance will be degraded if the sampling rate is too low.

In practice, the sampling rate is frequently recommended by the texts to be 20 times the bandwidth of the input signal being sampled [e.g. 3], because the system will very closely approximate a continuous system and most discrete system problems become insignificant. The absolute lower limit is set by the Nyquist sampling theorem described above, in combination with the need for adequate gain and phase information. The upper limit is defined by the controller delay, namely the program cycle time, the relationship to the word length of the computer hardware, and whether extended precision arithmetic is employed. If the word length is inadequate, the sampled signal will be sensitive to the effects of quantization.

The use of anti-aliasing filters on signal inputs has also been referred to. Such a filter is usually recommended as having a cutoff frequency five times the system bandwidth required [3] (the cutoff frequency for a first order filter is the -3 dB point, at which the phase difference will be 45°). However, the effect of filters on the overall closed-loop behaviour must be evaluated, particularly in terms of phase and gain effects. Similarly, output filters can be used but their effect on the overall behaviour must be considered.

7.2.2.4 Quantization

Quantization occurs because a digitized signal must be represented in a finite number of bits. It is rare that a number from the A/D will have an exact binary representation, and it must be either truncated or rounded. However, such finite precision arithmetic gives rise to non-linearities. These non-linearities can cause periodic oscillations in the output, even with a zero or constant input. These oscillations are called "limit cycles", and the range of values that the output amplitudes are confined to is known as the "dead band". The actual amplitude of the limit cycle is determined by the overall loop dynamics.

If white noise is added to the analogue input, it ensures quantization is a random process and eliminates limit cycle oscillations. This technique is called "dithering". Note that white noise is distinct from the noise frequencies that an anti-aliasing filter is used to remove. Its spectral content is uniform across the range, rather than concentrated at particular frequencies.

Dithering noise is not usually added in the digital stage as

- it may be coherent with the quantizer (and therefore not have the desired effect)
- it is often impractical (see [7, p. 185]).

7.2.2.5 Reconstruction of sampled signals

The simplest and virtually universal sampling method is called the zero order hold (ZOH), where the digitized signal value is held constant over the entire sampling interval. The signal can be recovered from the samples by using an interpolation formula, which for the ZOH is

sinc($\pi t/T$), where sinc x = (sin x)/x.

However, this method introduces a delay of half a sample period. This delay must be allowed for in the design of the control system. If the delay is not acceptable polynomial holds may be used instead, where the extrapolation function between sampling points is a polynomial, although these suffer from the disadvantage of requiring precision components. For example, in first order extrapolation a first order polynomial is used which is simply a straight line between the sampling points. Such a function is called a first order hold (FOH). A FOH requires knowledge of the next sampling point, which is usually extrapolated from the previous two [7].

Normally a ZOH combined with an anti-aliasing filter is used. Alternatively a low-pass filter may be inherent in the plant response. The advantages are

- simple hardware, without the need for precision components
- reliability.

The effect of the ZOH is best seen using a describing function calculation. The describing function of the ZOH is [8]:

$$C_1 = D\frac{\sin(\pi/n)}{\pi/n} \qquad \phi_1 = -\frac{\pi}{n} \tag{29}$$

where D is the magnitude of the input signal, C_1, ϕ_1 are the magnitude and phase respectively of the fundamental component of the output, and n is the number of samples per cycle of the input waveform (in effect $n = 2\pi/\omega T$, where ω is the bandwidth of the signal being sampled). The describing function is shown in graphical form in Figure 3. Note that the result is independent of the actual sampling rate and bandwidth, being dependent only on the number of samples, and is only valid for $n \geq 3$. In practice the usable range with a ZOH is ten samples or more [8, p. 412]. The effect of aliasing in reconstruction should be considered.

Pulse width modulation (PWM), where the pulses are of a fixed amplitude and the information is carried by the varying width of the pulses, is often used for output of control signals. For example, it may be used for variable speed control of a direct current electric motor with constant voltage source. The inertia of the motor and the load apply a low pass filter to the sampled signal to give a virtually constant speed. If PWM is being used, the pulse repetition rate should be many times the sampling rate. Note that PWM can introduce a variable phase delay.

7.2.2.6 Response times

In a real-time system the length of time required for processes such as the analogue to digital conversion must be taken into account. There is often a trade-off between response times and accuracy. Therefore, care must be taken when designing the system to ensure that the

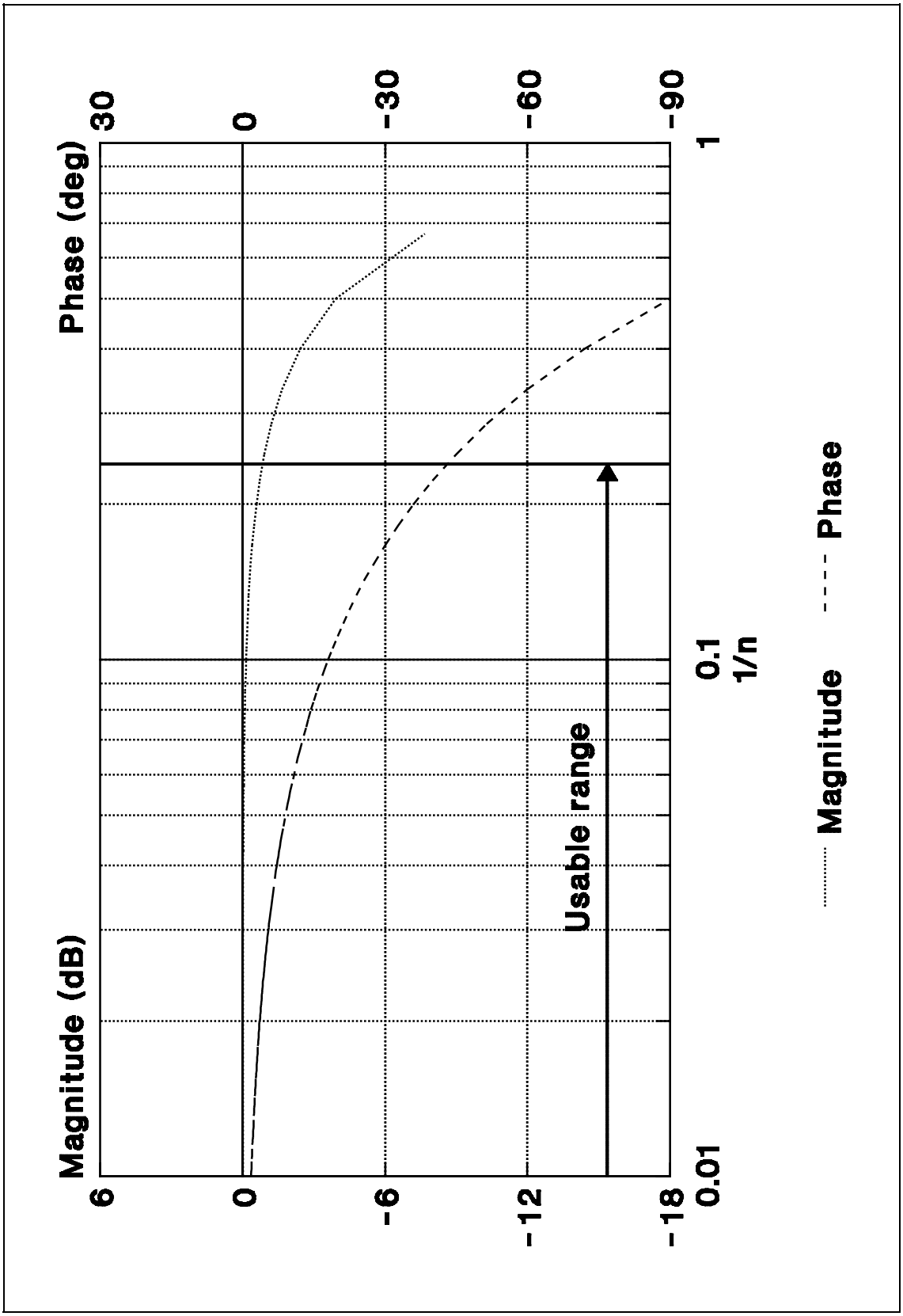


Figure 3 ZOH describing function

conversion has sufficient resolution (to avoid aliasing, for instance) but does not incur a prohibitive response-time overhead. Processing delays usually far exceed conversion delays, however. The implications of the sampling method chosen on the computational overhead must be considered. Generally, the programme cycle time (which is the effective sampling rate for the describing function) is equal to, or greater than, the sum of the conversion time and the processing time. The programme cycle time must be less than the sampling rate chosen.

Implementation issues are also important, for example in choice of the hardware, floating point versus fixed point and design of filters, as they will affect the processing time and the resolution. Many of these topics are covered in other MISRA Reports.

7.3 Conclusions

Computers offer significant advantages in the implementation of control systems. However, in describing a continuous process with a discrete representation a number of important issues must be considered:

- sampling rate high enough to avoid aliasing and reduce response times
- if sampling rate is software controlled, software timing should not be used so that jitter is avoided
- resolution and system response times must both be adequate, and should be designed against the requirements specification
- effect of quantization considered (see §14) and dithering used if necessary
- adequate design precautions to reduce the effects of noise signals, e.g. anti-aliasing filters
- evaluate effects on closed-loop control of
  - input filters
  - output filters
  - plant response.

8. The role of open-loop

8.1 Relationship between open-loop and closed-loop

Within the range and diversity of control techniques available to the systems designer, the relationship between open-loop and closed-loop is of particular importance.

Where there is both the requirement and the capability to adopt closed-loop control using established theory and techniques, the need to also incorporate open-loop mechanisms may appear to be minimal or even unnecessary. However, a variety of circumstances can contribute to the desirability of using both open and closed-loop methods in many applications. Part of the control system designers task is to establish an optimum balance and

separation in the design, development and operational resources used between the two techniques.

Examples of the situations where open-loop enhances and compliments closed-loop are:

- power up initialization, for initial optimum positions
- on re-initialization of the processor
- on detection of loss of adaptive parameters
- on state transitions
- fallback modes, on detection of sensor failures
- to improve transient performance, where a large transport lag exists
- open-loop computation of a closed-loop set-point.

8.2 Recommendations

An appropriate combination of open-loop and closed-loop can provide not only optimum control but also both redundancy and diversity, but this benefit can only be obtained if adequate separation exists.

- Incorporate a combination of open and closed-loop control.

- Use past experience and modelling techniques to determine the best balance.

- Provide clear separation between the two techniques in documentation and in the grouping of parameters.

- Use the difference information between the two techniques as part of "conditioning monitoring" and as an indicator of wear and deterioration.

8.3 Mapping of open-loop systems

Open-loop control of progressive, or infinitely variable systems is frequently parameter-driven with data, that is read-only in production, to achieve the optimum performance characteristics. This data, and the process of its optimization, often called mapping or calibration, is as important as the control program in ensuring that the system achieves the functional goals and is reliable under all conditions. In parallel with the development of the control systems and software engineering methods, specialist techniques are emerging for the development of optimum calibration data.

Of these techniques, a neural network represents one extreme form of a mapped system. Here the experience and principles of computer-based pattern recognition are used to "learn" the mapping requirements with a minimal need to understand the relationships between inputs and outputs (see §15.4).

Another extreme form of mapping is associated with the compensation of manufacturing variations by calibrating individual components at the end of the production line.

There are major benefits in maximizing the separation of the program from data, with few disadvantages.

8.3.1 The benefits

- Data is easier to change than a program.
- The analysis of non-linear effects and definition of algebraic expressions to cater for all situations can be very time consuming. Frequently, "trim values" are still required to compensate for unknowns; trim values may even be manufacturing specific and relate to one component.
- One software system configured with data can support many hardware variants.
- The application specialist engineers undertaking the task of calibration optimization need only a knowledge of the control methods used, rather than an understanding of the software engineering techniques involved.
- Separation of data from program is a form of software modularity and contributes towards the reduction of complexity.
- The production-intent control system and program code receives extensive real-world exposure during the calibration process. This can be a significant contribution to software testing and system validation, as a diversity factor is introduced by personnel of different engineering disciplines.

8.3.2 The disadvantages

- The separation of program and data may increase the memory and CPU resource requirements.
- The calibration optimization process may not adequately accommodate all factors associated with component manufacturing spread, tolerance stack-up, wear or other deterioration, in the way that a closed-loop system would.
- If the calibration data is considered to be a variable, then from a software validation perspective, the number of potential test combinations is increased.
- Since data are not hardwired into the controller, the code and system must be robust in coping with dtaa specification errors. If error detection is based on support tools capturing errors, then the tools must be verified to the same integrity level as the controller itself.

8.3.3 Recommendations

· Create a "user-friendly" environment for mapping and calibration, using the following:

- use a natural language interface for menus and error messages, etc.
- use appropriate engineering units with decimal numbers
- provide simple and easy-to-use data display and editing man-machine interface
- provide on-line help facilities
- ensure that data item name, title, descriptive information and potential range is displayed with the numerical values
- provide graphical presentations for two- and three-dimensional functions
- provide automated mechanisms for the transfer of datasets between revisions of program and hardware configurations involving different personnel to avoid manual re-entry of data.

· To ensure accurate, optimum results:

- use controlled and repeatable environments to collect and analyse data, and evaluate calibrations. This could be an engine dynamometer, a chassis rolling road or a computer model, etc.
- where appropriate use computer modelling to analyse the effects of wear and other deteriorations
- use experienced staff for the mapping and calibration activities
- generate calibration guidance notes that document the process of mapping and calibration. These guidance notes should bring together the experiences of the control systems specialists, software engineers, calibration engineers and other application specialists
- ensure that the calibration process is based on a spread of hardware components and statistical techniques that adequately accommodates factors associated with component manufacturing spread, tolerance stack-up and wear and deterioration. An alternative would be to determine calibration parameters that are specific to individual manufactured components
- assign engineering responsibilities to data and group the data items in-line with product features, the development organization, suppliers, etc., to enhance modularity. (See MISRA Report 7 [9]).
- use configuration management techniques, traceability, change and version control to data, and groups of data. This should be separate from, but compatible with, and cross-referenced to, the techniques applied to the control of the program
- conduct regular and independent reviews of calibration data
- where calibration data is used to compensate individually for manufacturing variation, there must be specific identification techniques and procedures to ensure that the data maintains a life-long associated with the component
- validate control programs and tools before calibration/mapping commences. Then validate the integrated system at completion of calibration.

9. Conclusions

Control theory is a vast subject which may only be summarized in a document of this type.

Above all, to apply feedback control as part of a software controlled vehicle system the design must be correct. A thorough assessment of the stability of the system must be carried out and documented. The documentation should include, but not be limited to

- the state equations of the system
- assessment of stability, including
  - choice of method and justification
  - the results of the assessment
  - any assumptions made
  - any qualifying statements required.

It is strongly recommended that standard texts are consulted for detailed discussion of the theoretical part of this document. However, experience forms a major part of control system expertise, and the recommendation must also be made to make use of the services of experienced control engineers to complement and guide those of software engineers in the design and implementation of software-based control systems.

Many systems use a combination of control types, including open-loop control. Careful consideration must be given to how best to achieve optimum performance, and also how best to implement each element of the control scheme.

Care must be taken to ensure that all parameters are thoroughly understood, and steps taken to ensure integrity to the level necessary to achieve the desired performance and reliability at each stage of the design process.

Part II — Design considerations

10. Human considerations

The most important factor when considering any system is the role of the human beings in the design, development and use of the system.

It must be remembered that the final product will be used by human beings. In the event that a fault is detected in a system, the resultant recovery action by the system must not detract from the driver's ability to control the vehicle; for example, a fuelling error which would render the vehicle's gaseous emissions outside legal limits must not result in the vehicle being disabled; or a sensor failure must not result in sudden, dramatic loss of power. Warnings of system problems to the driver must aim to inform and guide, not scare or dictate.

In the design process, the role of the human being is likely to be a source of error when functioning in a disciplined, logical fashion; human beings have a tendency to function in an undisciplined or illogical fashion when permitted to do so, with a resulting dramatic increase in the likelihood of creating errors especially in software.

The area of activity where errors are most frequently found is in the design of the requirements specification. It is common to recommend the use of a requirements specification writing tool or methodology, such as CORE; however, such a tool must be viewed as only attempting to impose discipline on human beings who use it; it cannot, of itself, eliminate human errors. The same is true of other tools; it must be recognized when applying tools or methodologies that they still depend on human input, and their value may be much reduced by human indiscipline or error. This is particularly true of version and change control tools; no matter what tools are in use, human beings will probably evade using them at some point. There should be a "safety net" procedure in place to supplement any tools to ensure that deviations from procedures imposed by the tools do not appear on vehicles "in the field".

Whatever methodology is used, it is essential for the minimization of errors to ensure that the human interface to any tools or software design system must be good; if it is not good enough, it is likely that tools will be used to less than optimum advantage, and the benefits which might be possible are not realized.

It is very important that engineers designing feedback systems should be of the appropriate background in education, training and experience. This should be interpreted as being control engineers, rather than software engineers, for instance for compiling the requirements specification. Whilst it is clear that software engineers must be employed to design and develop software, the availability of the expertise of control engineers will be very valuable in determining correctness of function and performance for a system employing closed-loop control algorithms to describe the primary function of the system.

11. The requirements specification

11.1 Functional and performance objectives

The first stage in the development lifecycle of a control system is the definition of functional requirements and performance expectations and objectives. Generally, techniques for specifying functional requirements are well established [12, 13], however, the specifics of a control system need special consideration and the expertise of practising control engineer appropriately educated in control is essential, even at this stage.

The performance attributes need to be quantified wherever possible and any predefined or "given" component identified. Effective modelling, as part of the design relies on numeric data that accurately describes the physical and dynamic characteristics of the object, or objects, to be controlled.

It is important to understand the scope of the requirements specification before committing to the detail of describing the parameters and functions; particularly in a large system, it is easy to lose sight of "what the system is about", and create anomalies, or errors of omission, by having a too localized or compartmentalized view of the system.

Examples of required performance attributes, with recommended dimensions and specification methods:

· Hardware and physical attributes

Physical nature of the object to be controlled	Drawing, size Weight
Controlled variable	Position, speed, etc.
Power,torque, load, momentum, etc.	kW, PS, Nm
Ranges of operation	Speeds, flows, etc.
Component tolerancing and wear characteristics	Transfer function, %
Variants and configurations	List
Controller physical characteristics, location and environment	Drawing, spec. ref.

· System attributes

Preemptive states	Names, definitions
Requirements for safe states	ditto
Setup and service modes	ditto
Transitional requirements between states and modes	Input combinations
Calibration parameter requirements, facilities and grouping	List, ranges, etc.
Relationship with other controllers	Document references

· Performance and accuracy requirements

Known impacting independent variables	List sources
Noise/jitter rejection	%
Steady state accuracy	% errors
Dynamic performance	Frequency response
Transient response times	Settling time
Stability/gain/phase margins	% overshoot (see §11.2)
Performance margin/robustness	

· Interfaces

Given sensor and actuator descriptions	Part numbers Transfer functions
Given sensor and actuator responses	Time constants
Communications interface specification to offboard diagnostics	Protocol spec.

11.2 Stability

A stable system is one that, when perturbed from a state of equilibrium, tends to return to that state of equilibrium. An unstable system may, when perturbed from a state of equilibrium, either tend to deviate further, or else tend to settle into a different state of equilibrium.

Stability requirements and attributes must be precisely defined, as part of the requirements specification. It is also essential that the technique for calculating stability forms part of the requirements specification, especially where part of the design process may be sub-contracted, and verification to the mathematical model may form a part of the acceptance procedures.

Stability margin is important, especially in self-tuning systems, or systems which employ error recovery or masking techniques. Stability may be difficult to predict with any confidence, especially in non-linear control systems. A variety of methods are reported in the text books for assessing stability:

- "describing function" method
- phase plane portrait method
- linearization (Lyapunov's first method): determines qualitative behaviour
- Lyapunov's second or direct method
- envelope methods: Popov and circle criteria

The Lyapunov (second) (or its inverse, Cataev) method is recommended to be used to predict stability, however, its use may result in unacceptably long convolution time. Mathematical sliding-control is now suggested as a way to improve stability and robustness; combined with fuzzy logic this has been shown to result in improved stability and convolution time. Application of a predictive algorithm to compensate for component response time (delayed

response, e.g. oxygen sensing) can both further improve robustness and improve control accuracy. The use of such mathematically complex constructs in a fast, real time system such as engine management will almost certainly require the use of a floating-point co-processor, in order to achieve the necessary computational accuracy. The use of a floating point processor may have implications for the integrity or testability of the system, however, and should be treated with caution.

Where there is significant lag in a system which may be critical to the system's operational integrity, it is important to acquire qualitative as well as quantitative information from sensors. This indicates that importance should be placed on the use of predictive algorithms, and use of methods such as Kalman filtering to improve system modelling.

Any default state resulting from error management is likely to influence stability, as is any error state. This is discussed further elsewhere in this document.

Attributes which may require modification to the stability of a system include:

engine management	cold starting "revving up" transient throttle response
ABS	yaw control
traction control	spin control
steering	use of rear steer in a transient mode to control steering characteristics transient steering response (swerving)
suspension	extreme bump or rebound condition anti dive tyre blow-out.

These are suggested as generic examples, and do not relate to actual systems.

11.2.1 Criticality analysis

A criticality analysis should aim to identify parameters which are critical to the system's stability and functionality. Sensors and actuators providing data from which parameters are derived, or acting on data from parameters, which are shown to be critical, must also be identified by the criticality analysis. FMEA should be carried out, and this should include quantification of effects on stability of various actions, including default states. Data flow and control flow paths should be identified. It is preferable to test for stability under as many fault/default conditions as possible. Simulation and emulation of the system should be used to facilitate early identification of sensitive parameters, or routines.

Boundaries of recovery should be designed and tested under simulation conditions to ensure that "domino" effects cannot take place.

Critical timing functions should be clearly identified.

11.3 Transient conditions

In order to achieve sufficiently fast response, stability margin is often reduced to an absolute minimum, and, under some circumstances, the system may be designed to become unstable or to operate in open-loop mode. (Aircraft are often placed into an unstable mode in order to achieve faster, tighter turns, for example; engine management systems in particular may be placed into either an unstable mode, or open-loop mode, under transient conditions to improve "feel".)

11.4 Formal specification

Applicability of formal specification methods to feedback systems demands careful consideration. It is conceded by the champions of formal specification methods that the cost of verification is much increased, and that the emphasis of software development cost is considerably changed. It is also often asserted that the use of formal specification symbologies both makes specifications very difficult to read to those untrained in their use, and that it makes application to other than small systems extremely cumbersome. The first assertion is undoubtedly correct, and with one exception, Larch, use of conventional word-processing systems is impossible. The second assertion is claimed to be untrue; however, since most automotive control systems may be considered small, possibly irrelevant.

Formal specification methods are split into two types: algebraic specification and model-based specification. The former, which includes Larch and OBJ, is built around object-orientation, and set and array theory; the latter, which includes Z and VDM, is function oriented. For use in control systems, the former appears the most appropriate, as its reliance on set and array theory particularly lends itself to control theory mathematics. It is important, however, to recognize that, if a control system has already been described by its mathematical design calculations (as opposed to trial and error), there can be no more formal and unambiguous description: formal specification will probably only offer any benefit in the areas of diagnostics and fallbacks. Formal specification of systems with concurrency is still in its infancy as yet.

There are other formal specification methodologies which may be considered as achieving the same objective: e.g. CORE, DEMOS, without the user-unfriendliness of schemes such as Z. Since the objective of a formal specification tool is to reduce errors, user-unfriendliness does not seem to be an appropriate attribute for such tools. However, one should not expect too much of such tools or methods: they still depend on the human writing the specification recognizing and including all state combinations correctly and unambiguously—they do not eliminate human error. If wrong premises are entered into the formal specification, or it is

incomplete, the product will be just as wrong—but perhaps with formal proof of its "wrongness" more visible.

11.5 Graceful degradation

A system should be designed to be capable of operating in a degraded state. If functions are adequately partitioned and errors managed , a single fault, error, or failure should not cause a system failure.

It should be possible to design a system in such a way that several tiers of degraded service are possible before system failure ensues.

11.6 Watchdogs and fault recovery

At the requirements specification stage, a decision has to be made regarding whether to opt for fault tolerant or fault masking design, and the general watchdog configuration specified. In either case, correctness of data should be checked throughout the program flow, and if an error is detected, recovery initiated.

The system should be partitioned so as to separate critical routines from non-critical routines, as well as from other critical routines. The consequence of each routine failing must be identified, and appropriate watchdog and default action specified.

Watchdog and default actions should be designed, and proved, to respond within timing requirements for critical routines.

When default states are being considered, the acceptable minimum level of prime service for the system must first be established; default states must then be proved to comply with that requirement. The result of more than one error being detected will result in more than one default state being invoked; the combinations of default states which permit this minimum acceptable level of service must be identified, and appropriate action defined for the event that this level of degradation will be exceeded. Default states, and state combinations, must be tested, and proved to comply with the laid down minimum level of service requirements, and also the system stability requirements.

The ideal fallback or reversionary control scheme for a safety critical system should be to safe, predictable and legal state. Failing that the system should revert to a safe state. The driver should be alerted to the illegal operation of the vehicle systems. All detected faults should be logged in a fault store (thus illegal operation of the system should be logged). A major difficulty is establishing the real cause of the abnormal inputs and logging this as the fault (in the past schemes have been implemented where a fault is logged, and then if it does not re-occur within a certain time period it is removed from the fault store; it is much better to target effort to determine the true fault and permanently log this).

11.6.1 Fault migration

Faults may migrate through a system by being passed from task to task, via inter-process communications. Fault tolerance may increase this possibility, by:

- failing to invoke a default process or value
- failing to limit a fault to the task in which it occurs
- forcing asynchronous action which causes a "deadlock".

Multi-processor systems are particularly prone to this "domino" effect, whether sharing common memory or via a network. Languages designed specifically for a multi-processor environment tend to force synchronism on the designer; these include Modula, Concurrent Pascal, Occam, and ADA. The latter is designed to be capable of handling either tightly or loosely coupled networks, and is often recommended for network application.

Control of fault migration is enhanced by strong partitioning between tasks, with error management designed into each task. Communications between tasks must be minimized; it should also be tightly defined and controlled, so that faults or errors may not migrate via unexpected routes. Processes should be grouped into recovery blocks, so that communication paths are taken into account when a recovery process is initiated, but the recovery process is restricted to only those tasks which are affected by the fault. Boundaries for the recovery block may be drawn statically, at the design stage, or dynamically; the latter, however, may cause a "domino" effect system failure, and should therefore be subject to a proving procedure.

Each time a fault recovery process is initiated, a test for recovery should be carried out, not merely an assumption of recovery.

11.7 Interrupts

The use of interrupts must be subject to a specific set of rules. Wherever possible, watchdogs should be polled rather than interrupt based, in order that the process of fault recovery is not inhibited by watchdog action. Where a watchdog is used as a time-out process, the period of time-out should be less than the critical time for any task which is crucial to the criticality of the system, and allow recovery within that time. If a watchdog is invoked in a multi-process system, where concurrency is involved, it must not act in such a way as to create a possible "deadlock"

12. Diagnostics

It is usually fairly easy to detect a complete failure of a sensor. The design will have placed the prevalent failure mode signals outside of the normal operational range expected.

Detection of gradual degradation of sensor data is a much more difficult problem (but is becoming a more relevant problem (via such documents as OBD II).

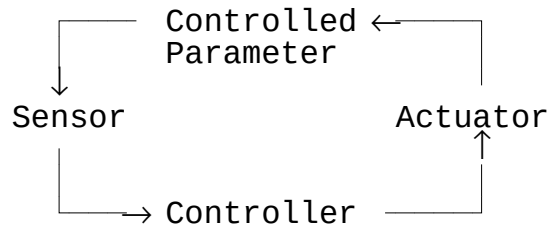
One method of detecting deterioration is to use an independent source of the same data. This may be a value read from another sensor on the vehicle (or evaluated from several other sensors via a model).

An alternative independent source could be test equipment employed during routine servicing (i.e. adopt a similar approach to wheel alignment procedures).

If a particular sensor is known (statistically) to have a significant degradation or drift after a certain elapsed time, running time or mileage then this period should be linked to a "service" indicator. It is debatable whether this sort of "failure" should indicate (and log) an "illegal" error as it is by no means certain that an illegal condition has arisen. This would appear to be a legislative problem (i.e. is it a legal requirement to have your vehicle maintained at the specified intervals by an authorized garage?) and is somewhat outside the scope of this document.

Gradual deterioration of a sensor may be detectable by long term drift of diagnostic data gathered at power-up but this depends on knowing the state of the vehicle and the environment (hot/cold engine, hot/cold day, etc.) which depending on the sensor may require a host of other sensor inputs (which have their own failure and degradation attributes…). The power-up state could be compared with a datum value set at the factory or the last approved service to determine if the sensor has become unusable.

Another diagnostic consideration concerns the state of the item under control. The control loop may be represented as:



The actuator (and elements of the item under control) will have parameters that the electronic controller is unable to directly influence or measure. For example, consider an hydraulically positioned mechanical element (such as an active suspension system). The controller may drive a coil in a servo valve. The controller may be able to sense the drive current and the position of the valve for "actuator local" diagnostics. The controller will be able to sense the reported position of the mechanical links of the system via the sensor (which may have "sensor local" diagnostics). If the integrity of the actuator and sensor is established then assessments of the state of the entire system can be made. For example, if the mechanical links do not move as fast as they should this can be an indication of increased friction (i.e. wear) in the system. To be able to make this assessment the controller needs to be continuously evaluating a model of the expected behaviour of the system (therefore the

processing resource implications of this model need to be evaluated). The expected behaviour of the system will be derived from a combination of the theoretical behaviour of the system (determined during modelling activities) and the long-term historical behaviour of this particular system (to allow for production variation of system components).

Sensors and actuators need to be health checked; this is a difficult function to carry out in a closed-loop system, especially where long term effects such as wear and contamination are involved. A health check of all sensors and actuators should be carried out at shutdown of the system, when an orderly shutdown sequence should be initiated. It may be possible to carry out such a check at start-up, however, in the case of, for example, an engine management system, the time available will depend on how quickly the driver turns the key from ignition "on" to "start"; once running, checks are difficult, though, in the case of an engine management system, if it is run in open-loop mode for a period after start-up, decisions may be taken by comparing data from various sources, and identifying "unreasonable" data.

Actuators may be harder to health check. In the case of solenoids, a negative-going step voltage of accurately controlled dimensions may be applied to the solenoid, and the back EMF measured (assuming there is no "catching" diode); however, there is a strong case for "smart" sensors and actuators which are self-monitoring, with a data reporting/default signal line separately provided for each sensor/actuator. "Smart" sensors and actuators may have the ability to be self checking even when the system is operating normally, in closed-loop mode.

13. Software security

Reliable operation of a system can be compromised by unauthorized access to, and modification of the system software. OBD II legislation in the US (California Air Resources Board) requires that steps are taken to ensure that tampering is either inhibited, or that clear evidence of tampering is left behind after unauthorized access to engine management systems.

Methods available to deter tampering are:

- passive
  - Sealing enclosure screws
  - soldering in/gluing in of ROMs
  - encapsulation.
- active
  - password access
  - checksum on ROM data
  - encryption of data.

The latter is only effective if the key to the encryption is stored in on-chip ROM in the processor, and therefore difficult to access. End-of-line programming can also be used to

render tampering difficult.

These comments are concerned with "one box" systems. Where the control system encompasses several communicating black boxes it may be necessary to demonstrate that the communications cannot be "hijacked" such that modified messages are not passed on to the control system undetected.

Automotive designs are very cost conscious. Provision of extra software (and thus faster processors) or extra hardware for encryption/decryption purposes must be designed in from the start.

14. Accuracy and scaling

14.1 Accuracy

14.1.1 Factors influencing quality of measured data

14.1.1.1 Dynamic range

It is unwise to utilize the total span of a sensor, particularly where there is a possibility of e.g. pressure spikes, in the case of a pressure sensor; it is equally unwise to utilize too little of the span, particularly where the section of span utilized is close to the minimum or maximum range of the sensor. It is not untypical for a sensor specification to recommend maintaining signals within the 10–90% span region. The bigger problem, however, may arise when the signal level is chosen such that conversion of analogue to digital quantities is predominantly using only a small proportion of the dynamic range of the A-D converter, (i.e. near to minimum). This situation may arise when several signals from sensors are multiplexed into one A-D, with fixed range input, and poorly chosen sensor characteristics for some parameters. Such a situation can result in inaccuracy due to both quantization errors, and noise entering, or inherent in, the circuitry used for conversion.

The same may also be true where a sensor offers low output signal level, and is compensated for by amplification prior to conversion. The signal may then contain significant noise, either due to circuit effects, or through disturbing signals entering the input cabling to the amplifier, or the amplifier itself.

It is recommended that sensor outputs are chosen such that as much of the output signal dynamic range is used as practicable, and that the dynamic range of analogue to digital conversion circuitry is utilized to optimum advantage.

14.1.1.2 Linearity

Account must be taken of the fact that sensors often have nonlinear characteristics; a good example is the oxygen sensor in catalysed vehicles.

Nonlinearity may be compensated for either in hardware or in software. In the latter case, a typical method is to use a signal input to calculate an offset into a look-up table, and use the nearest value from the look-up table, also, possibly, to calculate an interpolation between two adjacent values in the look-up table, if sufficient accuracy cannot be achieved from a look-up table alone. A constraint on this may be timing: accuracy may have to be sacrificed to some extent if the response time required of the system will not allow sufficient time for more detailed calculation.

Linearity may also be compromised if the recommendations of §14.1.1.1 are ignored: nominally linear A-D converters may offer unacceptable nonlinearity at the extremes of their input range.

14.1.1.3 Conversion time

In fast, real time systems, conversion and settling time of analogue to digital converters must be considered. If inadequate conversion and settling time is provided, errors may arise due to incomplete or incorrect digitization.

If a control process requires a given response time, performance of A-D converters must be chosen to allow adequate conversion and settling time without compromising the control algorithm; if, however, the A-D specified is too slow, yet adequate time is allowed for its level of performance, then errors may ensue, because:

- the measured value may differ from the real value at the point in time where it is used in the control algorithm

- time taken to service A-D conversion of input signals may reduce the time for the main control function such that the control algorithm is compromised in its ability to maintain correct control of the process it is carrying out. This may cause stability or accuracy to be affected.

The need for precision in the A-D process is worth considering carefully. Is it really necessary to use a 16-bit A-D, for example? Would it be more economical to employ an 8-bit A-D and an interpolation routine? It is also important to consider A-D linearity: a 16-bit A-D may offer poorer linearity than one of, say, 10 bits; the question to be asked is, should one accept a known, but reliable, lack of resolution, or better resolution with lower, and variable, accuracy? This question may be particularly important where cost is a major consideration.

14.1.1.4 Response time

There may be a significant lag between demand and response to that demand; a good example is the oxygen sensor, responding to throttle demand. The very noticeable effect of a sudden increase in throttle demand is that control of emissions is compromised until the system settles into its new, relatively stable, state. A similar example may be drawn from ABS sensing.

A car ABS system must respond much more quickly than a truck ABS system; this is because there is significantly more mass, and therefore rotational inertia, in the truck wheel than in the car wheel. If a truck wheel has stopped rotating, it will clearly take significantly longer to return to its original rotational speed than a car wheel. This also means that a lag must be "built in" to the system software to prevent the situation that a wheel is perceived to have stopped rotating, pressure is reduced to that wheel cylinder, the wheel does not immediately start rotating again, so either more reduction of pressure is applied, or the algorithm assumes a much more serious "skid" and holds off the pressure for much longer than is necessary. This may result in increased stopping distance, or the driver pressing the pedal harder—maybe even to the point of overcoming the ABS control, thereby possibly creating an unsafe situation. The dynamics of the system must be carefully considered when designing a feedback control system to ensure that control is maintained at the optimum.

It is often the case—and truck ABS is a good example, again, that the loop time of the controller software is considerably less than the response time of either the system being controlled, or even, possibly of some of its sensors and actuators.

It is important to differentiate, however, between delayed response time due to the device being controlled, and due to lag in a sensor. The stratagems applied to coping with sensor lag may not be appropriate for coping with delayed response of the device being controlled.

14.1.1.5 Noise

Noise may be apparent in a circuit for a variety of reasons; what ever the reason, it may adversely affect accuracy, either by virtue of causing signal levels to fluctuate, or by causing erroneous digital values to be perceived by the controller. Noise may also be a feature of the parameter being measured, for example, manifold absolute pressure: each cylinder will cause a "pulse" in the manifold, which must be "smoothed" or averaged out either in hardware or software. Some noise—for example, modulation on an interfering radio frequency signal—may become rectified by the victim circuit, and modify the perceived voltage, or cause thresholds to be changed in a digital circuit, in either case causing an erroneous measurement.

"Smoothing" or averaging processes however, add to execution time, and this must be taken account of in designing the system, especially in a fast real-time application. Smoothing must also be designed to ensure that the fastest desired rate of signal change is not compromised.

In general, it is better to design circuitry to reject noise, or tolerate noise, rather than try to effect an alleviation of a noise-provoked problem later in the development cycle. By this means, also, the characteristics of any filtering or averaging can be taken into account when considering response or run time performance.

The "integration" term of PID control is inherently a smoothing or filtering term, and may be sufficient noise reduction for this type of control.

Related topics: "Sampling and aliasing" (§7), MISRA Report 3 [10].

14.1.1.6 Damping

Damping and stability have already been identified in section 3 as of paramount importance when considering a feedback system. They can also be critical to accuracy; when there is either excessive under or overshoot, erroneous values may be measured, depending on the timing of the measurement, and it may be possible to cause or exacerbate instability by this means. Overshoot, in particular, may cause errors when parameters are calculated using output values for reference before the system has had time to recover from overshoot. An overdamped characteristic could mean that the object being controlled has not settled to the demanded state at the point where its output is used to calculate another parameter.

Any critical states such as those described above should be identified at the design stage if possible.

The discussion in the following section (§14.1.1.7) clarifies some of the reasoning in the above, as well as suggesting possible solutions to measurement problems.

14.1.1.7 Arithmetic systems

There are two basic schemes: fixed point/integer, and floating point arithmetic. A variation on fixed point is artificial or implied point.

The accuracy of all arithmetic systems is principally dependent on the word size for representing numbers. This means that for a given scheme, maximum accuracy will be achieved by a system which uses most of the bits of the words representing numbers for actual representation of the numbers, rather than for sign or decimal point recording. By definition, this means that maximum simple accuracy would be achieved by integer calculations, providing the integer values are near to the maximum word size. This however, is not usually practical, as in most automotive applications a representation of fractional values is required, and signal values may be widely varying variables anyway. This is most often achieved by using fixed point arithmetic.

Fixed point arithmetic has a significant shortcoming: if it is required to manipulate two numbers of vastly different magnitude, a very large word size is needed (e.g. multiplying 4086 and 0.00324). A way in which the size of words may be restricted is by the use of implied or artificial binary points. In an implied scheme, numbers are manipulated as integers, with the position of the point pre-calculated separately as a number of shifts, such that the decimal point is positioned in the expected place to suit the result. The accuracy of this approach may still be poor where the numbers manipulated are dramatically different, since a large word will still be required to represent the result—which may be impractical. Dynamic range depends on the word size, independent of the decimal point position. There are two problem areas with fixed point arithmetic: multiplication and division. In the case of addition/subtraction, the result has the same number of places after the point as the larger of the two decimal portions; when multiplying, however, the resultant decimal portion is the product of the two numbers of decimal portion places, and the resultant integer portion is equal to the product of the two integer portion places, plus one (for overflow indication).

This normally dictates that the number of decimal places in the result must be truncated or rounded to fit into the available word size. Division is similarly difficult, especially when dividing by numbers very close to zero. It is essential when dividing to use correctly preconditioned arguments.

The alternative is to use floating point arithmetic, which automatically keeps track of the position of the point. However, for a given word size, floating point is less accurate than fixed point arithmetic, owing to the need to store exponent values as part of the word for each number. The accuracy of floating point arithmetic systems is still dependent on word size, and though it may be much more readily able to handle calculations involving numbers of dramatically different size, the result may still not be of sufficient accuracy.

See also §14.2, MISRA Report 3 [10].

14.1.1.8 Accumulation of errors and rounding

It is commonly recognized that significant errors may occur in any system due to accumulation of errors and rounding.

Accumulation of errors due to tolerancing may be prevented by good use of scaling, and by care in design.

Accumulation of errors due to rounding requires more care. Prior to any manipulation of numbers scaling must be attended to, and word sizes chosen such that rounding is carried out so as to ensure an adequate level of accuracy. In general, it may be advisable to ensure that word sizes are large enough that rounding may be performed on the basis of the last two digits, rather than the last digit alone. Rounding values may also be stored and used at a later stage of the program to check and correct if necessary for accumulated errors.

Additionally, a particular problem can exist with fixed point arithmetic in a feedback system, when multiplying. The necessary rounding of the result will introduce an error which may nullify the effect of biasing error in the feedback loop; this is a particularly serious problem when integration follows multiplication, or when another multiplication follows, and the representation of the number must be modified in order to be used in the next process within the loop.

Two separate, and different calculation methods and paths may be used to provide an error checking mechanism for computation routines. One method may be a rough approximation method, used to give a guide for comparison with the other, detailed, method.

14.1.2 Factors relating to inability to measure

14.1.2.1 Inferred variables

Some input variables may be either difficult or impossible to measure accurately. This may be for a variety of reasons, such as the simple practicality, of measuring actual gas

composition of the exhaust gases; the latter is inferred by making use of the cooling effect of oxygen in the exhaust gases on a zirconia element sensor. In practice, there is also an inference employed in using the oxygen sensor in the feedback loop; combustion efficiency is related to air/fuel ratio in the control algorithm, by inferring a set of conditions defined by inputs from the oxygen sensor, the throttle demand potentiometer, the measured air flow into the engine, and engine speed. The latter may be further inferred by using a manifold absolute pressure sensor, and relating this parameter to air flow or throttle demand. Clearly, other parameters may be inferred from the input signals used, e.g. fuel type; a throttle potentiometer setting may be inferred by referencing mass air flow sensor and engine speed.

There are potential problems in this approach, however. A sudden opening of the throttle is not simultaneously accompanied by a proportional increase in engine speed; nor is it simultaneously accompanied by a proportional change in oxygen sensor output. It is common practice in such cases to employ a predictive algorithm, in the case of the oxygen sensor, for example, to estimate what its output will be when the engine achieves the operating conditions demanded of it. By this means, less deviation of air/fuel ratio from stochiometric may be achieved in the transient situation; in other words, accuracy of the system is enhanced. Since mixture inaccuracy is due to stability of the system being compromised by the response time of the controlled object—the engine—and the oxygen sensor, the predictive process may also be deemed to enhance stability.

It is often the case with present-day systems to revert to open-loop control during transient states; in this situation, accuracy of the mixture, and therefore, emissions, is totally reliant on estimated parameters designed into look-up tables in the controller. Clearly, this cannot often be expected to be more accurate than the predictive process—although of course, the latter will be subject to the possibility of error. The open-loop transient case may also be considered to be a variant of inferred parameterization.

14.1.2.2 Statistical estimates

A parallel method, or adjunct, to inferred states is statistical estimation. In this case, a number of samples may be taken of a parameter over a set period of time, and a statistical estimate taken of their probable "real" value. This is quite different to averaging, where a number of samples is taken and the mean calculated; in the statistical case, a typical approach is to calculate a probable value based on a gaussian or Weibull distribution.

The parameter required may be directly measured, for example in the case of a noisy signal, or may be inferred from examination of other related parameters. This method may have advantage where the desired parameter is affected by slow response or settling time, as well as by noise.

The techniques of inferred variable and statistical estimate may be combined; however, as with all such processes, there may be dangers in becoming too far removed from the "real world" signal, and caution is advisable. Such techniques must not be used as a substitute for good circuit design, or to compensate for poor mechanical design in the device to be controlled.

Such techniques may be used in the internal diagnostics software for assessing condition of sensors, or in the definition of default states.

14.1.2.3 Artificial variables

This topic is slightly more removed from the "real world" than inferred variables. In the case of artificial variables, these are values derived purely from theoretical calculations based on data from output states, or simple measured data from other, non-related sensors. This process is often operated successfully in industrial automation processes; in vehicles, it is only likely to be used for default states in the event of a component failure—or more probably, multiple failure. It should not be used in situations where it can directly impact safety, unless it can be proved that the effect on safety is so negligible as to be non-existent.

14.1.2.4 Observers

An important concept is that of observability, which is similar to controllability and related to it by the control transfer functions. This subject is complex, and cannot be adequately described here. A fairly detailed mathematical description is contained in Reference (2), Chapter 5. Control system state variables have been asserted by Kalman to fall into one of four categories:

- controllable and observable
- controllable but not observable
- observable but not controllable
- neither controllable nor observable.

A control process is defined as controllable when all of its output states can be set by manipulation of any or all of its inputs; it is observable when all of its input states may be derived from any of its output states. If any input or output state cannot be derived, the system may be uncontrollable or unobservable. The concept of observability is of particular importance to the specifics of accuracy.

By definition, the solution of any control equation matrix is a difference equation matrix. Accuracy may depend on using this difference equation to implement a further degree of freedom in the control process—to calculate a correction value to be applied to the output. An observer is an implementation of this difference equation matrix to either estimate inaccessible parameters, or to improve control accuracy. An important implementation of the observer is the Kalman filter.

14.1.2.5 Kalman filters

In industrial automation, Kalman filters are used to provide estimates of unmeasurable process variables, on line, and also to process inaccurate measurements to improve accuracy. In effect, the Kalman filter is a technique based on forming a parallel feedback loop which will output an error value for any state of the main process which it is observing; this value may be used to apply a correction to the main algorithm, or the Kalman filter may be used as an

"on-line" correction mechanism in its own right. The Kalman filter smooths and extrapolates the time series for the system; it requires a very good model of the system to be incorporated into its algorithm in order to effect good results.

A disadvantage of Kalman filters is that they impose a need for considerable additional computation, and usually, considerable development effort. In the case of linear control systems, this may be acceptable; for use in nonlinear systems there is very considerably more computation required than for the linear case: nonlinear systems must be treated as a series of small linear steps. However, Kalman filters are growing in usage in process control industries, and the methodology emerging from that source may in the foreseeable future be applicable to vehicle applications.

14.1.2.6 Feedforward control

There are three ways in which observers may be used to improve control accuracy: to calculate modifications to the control algorithm model, feedforward control, and predictive control.

Feedforward control is a process designed to calculate errors and apply correction on-line. This requires that the causes of error are independently measured and are applied to the control signal before the errors appear at the control output. This is the inverse of feedback control, where the errors are used to provoke the correction. A possible application of feedforward control could relate to the oxygen sensor during transient or rapidly changing engine operating conditions. To be successfully implemented, a very good model of the system states is required, also the control actions must be calculated very accurately.

14.1.2.7 Predictive control

Predictive control requires a good knowledge of expected actions, which may be based on either historical data or extrapolation algorithms. A very good example again is the oxygen sensor: it is relatively straightforward to predict that, if the driver presses the throttle pedal, the engine is expected to produce more power, more fuel will be supplied to the engine, and so on, therefore by storing historical data regarding engine states during throttle demand change of state, correction to be applied to the oxygen sensor input to the controller may be readily predicted.

14.1.2.8 Neural networks

Neural networks may be regarded as a form of observer, which "learn" the characteristics of an application, and then are able, having learned the characteristics, to take subjective decisions about the application. The learning process is essentially one of pattern recognition, and many iterations of the process are necessary, with a wide range of data, to enable the neural network to learn enough to establish a pattern.

Neural networks exhibit a similar decision taking process to human beings; issues which may not be well enough understood to define or convert into a computer program may be

implemented by a neural network algorithm. In some quarters, neural networks are viewed as being inherently safe—probably safer than conventional sequential logic programming; unfortunately, conceptualization of formal proof of a neural algorithm is currently beyond the ability of the implementors of such algorithms.

The neural algorithm is not dissimilar to Kalman filtering algorithms.

14.2 Scaling

Refer to section 14.1.1.7 (Arithmetic systems); MISRA Report 3 [10].

14.2.1 Reasons for the use of scaling

(a) The most common use is to match the output signal from a sensor to the input requirements of an A-D converter. There may be a multiplicity of different types of sensors in a vehicle electronic system, measuring a variety of parameters, and emitting outputs variously as voltages, AC or DC; current, AC or DC; pulsetrains, monopolar or bipolar, rectangular, sinusoidal, or of irregular shape. All of these signals must be conditioned in some way before being input to a system; after conversion to a digital signal, further scaling may then be required for the internal use of the digital signal by the microprocessor.

(b) A common need in vehicle sensors is for linearization. This is perhaps best exemplified by the oxygen (lambda) sensor, which, in its normal operating range is very nonlinear.

It may be possible for raw data to be employed, particularly if the data is to be used only for comparison with a fixed reference level. However, often measurements may be required which can be compromised by nonlinearity; nonlinearity may result in reduced discrimination in a critical range of operation, even where a simple reference comparison is concerned.

(c) A sensor output may form a critical part of a feedback path; by definition, a feedback path requires precise gain and phase control if it is not to court the possibility of unstable operation. The output of some sensors varies significantly with frequency, as well as varying from sensor to sensor in terms of gain versus frequency and phase versus frequency. In this situation, the scaling may need to be automated.

(d) Output signals may also need to be scaled either digitally or after conversion to an analogue value, in order to match the characteristics of an actuator.

(e) Another reason for the use of scaling is to condition numbers prior to arithmetical calculations. This is often necessary in fixed point or implied point arithmetic in order to optimize the use of the available word size. However, the accuracy

of arithmetical calculations should always be carefully considered, since, even with scaling, the need for a large word size in order to achieve the desired accuracy cannot easily be avoided.

14.2.2 Techniques for scaling

14.2.2.1 Hardware

In many cases, sensors may be employed with internal signal conditioning and possibly remote range adjustment; this can be a very convenient approach especially for low-volume systems, where the development cost of customized sensors cannot be justified. In this case, either sensors must be specified which offer a standard output scaling, or which may be programmed to a specific output range by the system they are serving.

Other sensors may be "signal conditioned" by amplification (or attenuation) within the host system; this may also involve differentiation or integration, with or without some form of amplitude clipping or levelling.

It should also be remembered that there may be advantage in considering mechanical scaling of a parameter: for example there may be a point where a mechanical quantity to be measured is at a more suitable amplitude, or may be inferred from a related parameter which presents a more reliable situation from an electronic point of view. The mechanical designer’s convenience may cost the electronic designer dearly.

14.2.2.2 Software

Optimal use of sensor dynamic range may be achieved by the use of auto-scaling algorithms; two possible approaches to this in closed-loop systems may be:

- use of sensors with calibration devices included in their design, which may be activated by the processor; having established the sensor span, the scaling may then be adjusted by comparison with "maximum operational limits" stored in ROM

- use of open-loop measurements to adjust scaling to the optimum for each sensor.

However, it is more usual for scaling to be set manually, and programmed into the system as look-up tables. The use of look-up tables also offers a convenient means for linearization, or offsetting a zero point, as well as adjusting "gain". If it is required to only set "gain" to set range for a sensor, this may be achieved by simple multiplication or division by the processor, of the digitized signal input.

It may be necessary, however, to carry out more detailed actions when setting scaling. There may be a need to consider diagnostics, in which case the value of input signal measured may be compared with other parameters related to it, and an inferred maximum value calculated

for comparison with the maximum value calculated from the actual measured signal. This information could be used to diagnose sensor damage such as a range shift which would result in severe nonlinearity or "clipping" at the maximum value of the mechanical parameter being monitored by the sensor.

14.2.3 Possible sources of scaling error

Sensors may suffer three significant sources of error:

- zero drift with ageing or temperature
- linearity change with ageing or temperature
- sensitivity change with ageing or temperature.

All three may be brought about by contamination or physical damage to the sensor. A classic case of this is overstressing of a pressure sensor diaphragm by an unexpected pressure spike (e.g. a "backfire" into an inlet manifold may overstress a manifold absolute pressure sensor), or poor attention to servicing or perhaps engine wear may result in chemical attack weakening the diaphragm of an oil pressure sensor. The only way of identifying such changes in sensors is through careful diagnostic procedures, in which the sensor is tested at a minimum of three points in its operating range. Simple range checks will not show up changes in sensitivity or linearity.

A zero drift over a long period of time may be tested for very simply; zero drift as a system"warms up" in use requires more frequent tests rather than only when the vehicle is started. At very least, sensor zeroes should be checked at start-up and switch off, and compared, if there may be unacceptable error brought about by zero drift. The presence of zero drift of greater than a set value may have diagnostic value in itself in commenting on sensor health.

Potentially, there are greater possibilities for error in systems which use hardware scaling techniques. Unless there is a suitable calibration arrangement, which is used in the diagnostic process within a system on an automatic, regular basis, there will be no direct indication that an error is present.

The use of pulsetrains to indicate engine or other speed or, for example, angular position, may also result in error. The accuracy which is required must be carefully considered and the sampling technique configured accordingly, where it is required to measure a range of speed from near zero to some quite large quantity. There may be particularly large ranges involved in the measurement of engine speed and wheel rotational speed for ABS. In the latter case, the minimum speed for ABS action to be maintained may have a significant impact the range needed to be measured, given that, where magnetic sensors are used (popular for this application) both pulse amplitude and shape may be significantly different at the extreme low speed end of the range. Combined with variations in air gap to the toothed wheel, and tolerances, there can be significant differences between sensor outputs from wheel to wheel, as well as vehicle to vehicle.

Positional measurement such as crankshaft angle for ignition/fuel timing may also be subject significant error, particularly as wear takes place on the crankshaft bearings. In particular, endfloat of the crankshaft may cause significant variations in the apparent position of the pulse signal. In diesel engines, the problem is far greater, since much greater precision is required than for petrol engines.

Some engines use camshaft angle rather than crankshaft angle; a further inaccuracy may be created by wear and stretch in the timing chain or belt.

15. Optimization and adaptive control

15.1 Introduction

What is the difference between optimization and adaptive control? Optimization is principally a "once-and-for-all" process, an open-loop technique, in which the human operator closes the loop, for each parameter being considered. It may also take the form of, for example, adjusting the "CO pot" of an engine management system at MoT time. Adaptation is essentially a closed-loop function designed as part of the control algorithm, carried out while the process is active, and constantly under review during operation of the system. There is no requirement for human intervention in the adaptive case. Other than this, the two processes are based on similar theory.

There are two constraints to be addressed for any control scheme:

- it is rare for theory to be borne out by practice: some "fine tuning" is virtually always necessary, and this is often a manual adjustment process
- where a large number of similar systems are manufactured, as often in the vehicle domain, there will inevitably be differences between systems, both in terms of device being controlled, and the controlling device. There is also the long term variation due to wear and contamination to consider.

15.2 Optimization

In order to carry out an optimization, it is necessary to decide:

- what is the objective of optimization?
- what parameters need to be adjusted in order to achieve the desired objective?
- what are the constraints on achieving the desired objective?
- how will optimization be achieved in practice?
- what measurements will be necessary to quantify or define optimization?

In practice, analysis shows that few control systems are linear; far more are only so nonlinear that a linear approximation produces adequate results. Nonlinear systems may be split into two types, those in which strong nonlinearity is an undesirable characteristic, and those in

which strong nonlinearity is dominant part of the system's behaviour. It may be possible to analyse some systems in which strong nonlinearity is undesirable, however, some of the characteristics in those systems in which strong linearity dominates may also be present in all nonlinear systems to some extent.

A particular problem present in nonlinear systems is that output of a strongly nonlinear system is (harmonically) displaced in the frequency domain, and the harmonic orders and amplitudes will usually be gain dependent.

This is shown by application of a sinusoidal signal to a non linear transfer function *f*. Consider a simple nonlinear feedback system decomposed into linear and nonlinear components, as shown in Figure 4.

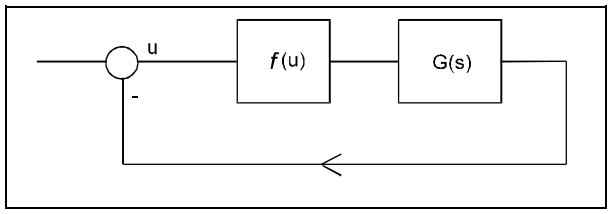


Figure 4 A simple nonlinear closed-loop controller

Applying a sinusoidal signal $u = a\sin\omega t$ to $f(u)$ gives

$$f(a\sin\omega t) = a_0 + \sum_{k=1}^{\infty} (a_k \cos k\omega t + b_k \sin k\omega t) \quad \textbf{(30)}$$

As well as the fundamental component, the output contains a DC shift and harmonic components. The expression above may be developed to provide the "equivalent gain", if the harmonic terms are ignored (as in the describing function method); however, it should be noticed that this term has real and imaginary components, therefore stability of the loop should be analysed. For this method to give satisfactory results, the harmonic terms must be relatively insignificant, or possibly neglected if the device being controlled is incapable of responding to them; this is often not the case. In the situation where the device being controlled is incapable of responding to the harmonic terms, it is still possible for the behaviour of the control loop to be defective if the harmonic terms cause saturation of part of the loop.

For second-order systems state planes may be employed, though these are principally used for relatively simple linear second-order systems; they become exceedingly complex for application to non linear systems, especially where there are a significant number of variables.

In principle, optimization is a process of finding the minimum value of the derivative of the cost function—assuming that the function has a maximum value. In a linear, non-recursive system:

$$\frac{\partial C}{\partial p_i} = 0 \qquad \forall\, i \tag{31}$$

where C = cost function; p_i = control parameter; i = ith order.

Alternatively, if there is no minimum cost value, then the maximum value of a closed interval must be sought. Where optimization depends on the relationship between two variables, the maximum value for their scalar values will be at the centre of a "contour map" relating the effects of the two parameters. Where more than two values are concerned, the "contour map" becomes a three or multi-dimensional entity, and derivation of the zero-value derivatives will be difficult, or may even become impossible, for significantly nonlinear systems. However, it is a simple task for linear non-recursive systems (but may be much more difficult for linear recursive systems). In nonlinear systems, there may be as many minima for the derivative of the gain function as there are parameters, or more.

The most significant problem of nonlinear control systems is that dynamic behaviour becomes a function of amplitude. This may be shown by examining the behaviour of a nonlinear system in the frequency domain: energy input at frequency A will result in energy output at frequency A plus a series of harmonic order frequencies each of which will have an amplitude dependent on the characteristics of the system. In the worst case, frequency A may not be present at all in the output spectrum.

Nonlinearity may take one or more of five additional specific characteristics:

- limit cycles
- jump resonances
- stick-slip motion
- backlash
- hysteresis.

Limit cycles are the characteristic action of "bang-bang" controllers; they also happen in nonlinear systems due to local oscillatory states. In this case, the system can never, if such a state is encountered, settle to equilibrium. "Bang-bang" or simple relay control is used in situations where maximum speed of response is required. It is usually necessary to prevent the inherent oscillatory behaviour of this method by provision of a "dead band" centred around the desired output state.

Limit cycles can be described in their stable oscillatory state by continuous nonlinear differential equations.

Jump resonance is caused by a saturation nonlinearity. In this case, saturation, of an amplifier within the control loop, should be relatively easily identified by investigation with conventional instrumentation.

"Stick-shift" motion is a problem related to the device being controlled; some feedback systems requiring precise control of a device which suffers this phenomenon employ application of a constant "dither" signal to the device to ensure that the stiction is minimized. The effect of "stick-shift" motion is to degrade system performance: the device being controlled will require a larger input to start it moving than it will require to achieve the amount of movement demanded. Effectively, this causes the device to overshoot relative to demand.

Backlash is similarly a mechanical problem in the device being controlled, making precise positioning difficult. Providing backlash is constant throughout the required operating range it can be compensated for in the control algorithm. It will, nevertheless, degrade the system performance.

Hysteresis is a similar problem, but in the electronic control system.

Leigh describes the problem of nonlinear system analysis by stating that linear systems obey the superposition theorem, whilst nonlinear systems do not. What this means is that techniques which are dependent on transfer functions are invalid (Laplace, pole-zero, root-locus, etc.) Step response functions also do not offer a solution, because of the amplitude dependence of nonlinear systems. It is rare for nonlinear differential equations to offer a closed form solution in practical situations.

The problem with carrying out nonlinear control system analysis is principally one of complexity: in a system of any size, it is beyond human intelligence to create the necessary scale of mathematical interrelationships and interactions involved. The most often used approach is to develop a rough approximation based on linear models for a range of states, and develop from that to a usable system by trial and error.

In practice, for a complex system such as engine management, the system variables need careful consideration to identify the parameters critical to achieving the desired system optimization objective. The objective may be: NOx less than "Z" ppm; CO less than "Y"%, power more than "X" bhp; specific fuel consumption less than "W" gm/kWH. Some of these objectives will conflict, therefore a compromise will be unavoidable. That compromise will inevitably be dictated by the output variables as much as the input variables. Some output variables may indeed be mandatory, e.g. CO and NOx maxima, therefore, other parameters may have to bear the brunt of compromise. It is more often than not the case with such a complex system as engine management that the optimization will be performed by trial and error, utilizing to the full the skill of the calibration engineer. That this process is difficult to define or formalize is no greater criticism than that attached to the severe difficulty of developing an optimization by mathematical methods for such a complex, nonlinear system.

There may well be points in the variables map for a system such as engine management where limit cycles occur, or other discontinuous nonlinearities, which force a need for solution by trial and error or manual manipulation of the mathematics. In this event, the solution arrived at will probably defy theoretical analysis.

It is a significant help to optimization if the number of parameters used to adjust to the desired output is minimized: even in an engine management system, despite the large number of measured parameters, very few have a significant enough effect to be useful for optimization.

15.3 Adaptive control

Adaptive control is effectively on-line optimization. The adaptation algorithm for a nonlinear system may become very complex indeed, to the point of unrealizability. It is important to restrict the numbers of variables used in the adaptation algorithm to a minimum for achieving the required result (i.e. minimize the degrees of freedom). This principle is clearly seen in the case of engine management systems, where most adaptation is based on the lambda sensor value. This value must be maintained within an acceptable range of values; when it falls outside an acceptable range, adjustment is made to the reference point in the fuelling map until equanimity is restored. If the correction to the fuelling exceeds predefined bounds, the driver is given a warning that the system needs attention (OBD II).

The use of look-up tables for this type of adaptation is a way of compensating for the many nonlinearities in the system: the adaptation is carried out purely on the basis of the output of the lambda sensor (which is itself highly nonlinear), and effectively represents a simple control loop, independent of the main control loop.

15.4 Neural networks

Perhaps the optimal form of optimization is offered by neural networks. In principle, a neural network is a self-learning system, which may employ additional feedback in the learning process or on-line, if desired.

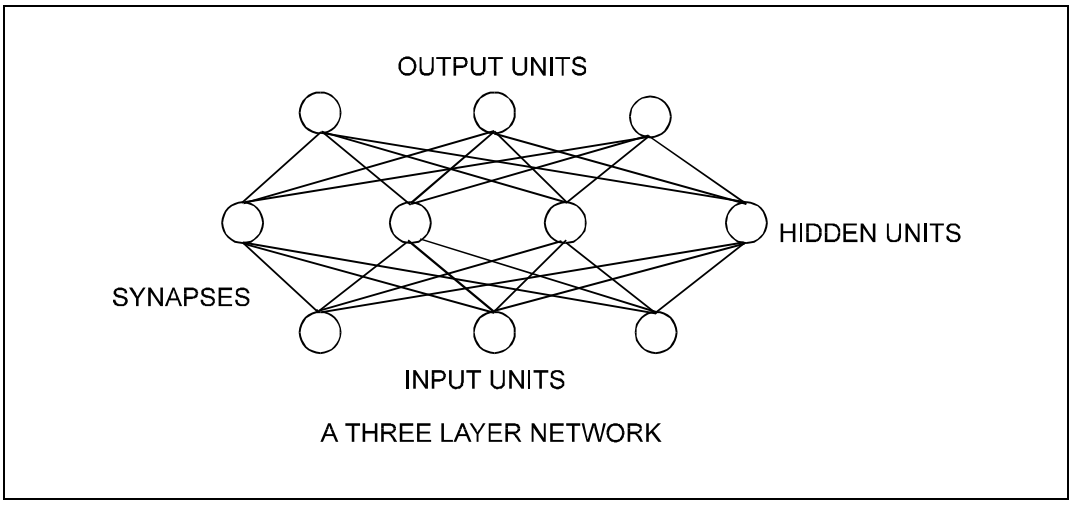


Figure 5 The multilayer perceptron

The basic neural network is called a multilayer perceptron (MLP); it is illustrated in Figure 5. Each unit or node is represented by a piece of processing power, each synapse by a transfer function or "weight". The principle of the MLP is to minimize the sum-of-squares error with weights determined from the data set:

$$E = \sum_{(\text{patterns})} \sum_{(\text{outputs})} (y - t)^2 \qquad \textbf{(32)}$$

where y = output, t = target.

Error backpropagation is used to evaluate $\nabla_w E$ (the synapse weighting function), using gradient descent, conjugate gradients, etc.

It can cope with nonlinear terms, and may cope with a very significant number of parameters, which may be individually weighted. However, there are two significant problems for neural networks:

- the quality of the result depends on the quantity and integrity of the data used during the learning process
- validating the process is comparable to validating a human being's learning processes.

It is essential for data to be consistent: the principle of the neural network is "if *X* and *Y* happen consistently, the output should consistently be *Z*". If however, *X* and *Y* sometimes imply an output of *B*, or *X* and *F* sometimes imply an output of *Z*, the neural network may not cope with it—just as a child in its learning process. There must also be an adequate spread within defined operating boundaries of values for *X*, *Y* and *Z* for the neural network to learn properly. Neural networks are very susceptible to the quality of data used in the learning process, and the latter must be expected to be a protracted business. Learning times of weeks or months are not unusual, with vast amounts of data being input to the neural system.

Validation presents a serious problem. It may be possible to validate all or part of the neural software and the hardware on which it is implemented as individual items; the correctness of the learning process is dependent on a variety of variables, at least some of which are essentially subjective. Subjective processes cannot be formally validated.

16. Diagnostics and fallback: predictive techniques

16.1 Introduction

As automotive control systems strive to achieve finer control over existing systems (e.g. emission control) or are applied to new (safety-critical) areas (e.g "brake by wire") their dependence on accurate measurement and control of the external world assumes more

importance.

Sensor installations (sensor, connectors, wiring) suffer from degradation as does the actuation path. Continued control of the system in the face of such degradation requires detection of the reduced performance of the instrumentation/actuation paths and some alternative method of generating the required sensor input and actuator drive. In aerospace and safety-critical industrial applications the use of multiple independent channels of input, processing and output is commonplace to ensure continued safe operation of the system; for cost reasons this approach is not really applicable to the automotive market. To cater for component failures the electronic control system should be able to easily detect gross changes in sensor or effector characteristics and provide enough control to bring the vehicle safely to a standstill.

However, some parameters that the electronic control system uses may not affect the immediate safety of the vehicle occupants or other road users (for example, exhaust emissions). OBD II requires that failures (or degradations) of any exhaust emission control related component are annunciated to the driver; the problem is to detect slow degradations of sensor or effector characteristics. Some of the possibilities for handling prediction (and accommodation) of degradations are discussed below.

16.2 Lifetime

The crudest way to avoid component degradation pushing system performance and integrity outside the operational envelope is to simply have a policy of replacing items that have been on the vehicle for a certain period of time. The "time" period may be elapsed time, vehicle operating time, distance travelled. A more sophisticated life calculation may factor engine running time by engine speed, for example. This approach is employed for standard service items (such as oil filters) and for indicating that a service is due.

If an automatic system is to be employed to alert the driver that a component requires replacement then a means must be provided to measure (and store) time and reset time counters for replaceable components. This implies the provision and management of non-volatile memory.

The service actions would get quite involved where items with different replacement rates are involved.

The calculation of replacement intervals is performed "off-line" (i.e. away from the vehicle). The vehicle manufacturer must determine the lifetime limits for each component whose degradation affects the safety (or legality) of the vehicle. Data contributing to these calculations will have been collected from proving sessions, service returns, i.e. the timeout limit will be set from a conglomeration of results pertaining to the operation of a sample of units in the past.

Rather than having a set time limit in which to replace a component it is more economical to replace a particular instance of a component when it has degraded. Factoring a fixed

lifetime by vehicle operating conditions is a step towards this but ideally we would wish to directly measure the parameters of a specific component.

The aim of the lifing calculations on sample units is to determine a common parameter limit which can be used to indicate impending failure. This may not be as simple as may first appear due to initial manufacturing spread on a component (less spread implies a higher cost component). What must be measured is not the absolute component performance but the current performance relative to the initial performance of the component (see "Trend Logging", below). To examine component performance one must excite the component in a standard way—for a sensor this implies that the component be subjected to a standard "real world" stimulus. A major difficulty is determining (independently) what the value of the real world stimulus is so that one can determine if the sensor is faulty—this is discussed below.

16.3 Multiple channels

One of the difficulties in assessing a component is having a standard to measure it against. If two sensors measuring the same parameter are available then one can be compared against the other and a failure (total or partial) of one sensor can be identified. To avoid degradation affects from compromising multiple sensors in the same way it would be advantageous to use dissimilar sensors to measure the same parameter.

Aerospace applications commonly use two or more independent sensor channels into the ECU. This approach is probably not viable in the cost sensitive automotive environment. However, it might be possible to employ other sensors in the vehicle providing that the sensor data can be transferred, i.e. the vehicle is multiplexed. Data from all sensor inputs could be collated by a central "body computer" and sensor degradation could be determined. In this scenario note that the control units are not (necessarily) accessing sensors over the network—this is probably a good idea from the safety integrity point of view—higher integrity systems should not rely on data from lower integrity systems for primary functions. The central body computer could store all the vehicle sensor history as well handling the lifing algorithms, pending component failure could be flagged to the driver and the ECUs using the data from that particular component so that a fall-back control strategy could be selected.

The degree to which such a strategy could be implemented depends upon the extent of multiplexing/networking within a vehicle. Some "non-multiplexed" vehicles may have local communications between, say, engine management ECU and gearbox ECU which (bandwidth permitting) could be employed.

16.4 Inference from other parameters

As an alternative to a second sensor providing a standard for comparison it may be possible to model the physical parameter that sensor "A" is measuring from the outputs of sensors "B" and "C". This approach has to be thought out very carefully as obviously the validity of

sensors "B" and "C" have to be established (somehow) before sensor "A" can be evaluated—care has to be taken that loops are not created where each sensor is validating the other.

16.5 Inference from startup conditions

A standard to compare component performance with could be derived from system startup conditions. For example, as part of the system power-up checks a valve could be driven to an end stop and then the associated position sensor could be read. This sort of test would show a fault in the ECU-effector-world-sensor-ECU loop although localizing the fault for diagnostic/repair purposes could be difficult (is the fault in the actuator or the sensor?).

The startup conditions for some parameters may be difficult to define as they could depend upon the short-term history of the vehicle (e.g. temperatures).

Extended system start-up times while the system performs extensive power-up diagnostics are probably unacceptable form the customer's point of view.

16.6 Trend logging

An approach to detecting slow degradation of a component is to record the performance of the component over an extended period of time. For example, the dynamic performance of an actuator could be logged; increased friction would show up as reduced acceleration and/or non-linearities in demand versus position. If the performance is getting consistently worse then the performance can be projected into the future and if unacceptable performance would occur in the short-term then a warning can be given to the driver.

The provision of (long-term) trend logging within an ECU implies non-volatile memory within the ECU (or somewhere on an internal vehicle network).

16.7 Servicing

Ultimately, a vehicle sensor can only be checked by comparison against a defined, maintained standard. The natural time for such a comparison to take place is at service time where test facilities external to the vehicle can be provided.

During servicing the state of the components on the vehicle can be read and compared against external standards, either manually or automatically.

The standards held at each service centre must be maintained and provision of automated test/diagnosis equipment will be expensive unless a standardized approach is taken (see MISRA Report 1 [11]).

One advantage of having formalized servicing is that product reliability and performance data can be collected and then transferred onwards to the manufacturer for collation in a centralized database.

16.8 Legal implications

Requirements such as OBD II demand that potential failures of components that affect certain features of the vehicle (i.e. emissions) must be detected and annunciated. If the only way to achieve this is to have lifetime monitoring backed up by sophisticated servicing checks this has an impact on cost of ownership of the vehicle. Cost of replacement parts may be mitigated by re-calibration of the system to use a degraded (but still acceptable) component.

Implementation of the fault prediction facility on the vehicle itself may not be possible to the accuracy available with external equipment; in this instance the on-vehicle system can only advise that the driver have a full check performed at a suitable (approved) service centre. If drivers are continually returning vehicles for full checks and no faults are found this will reduce customer confidence in the vehicle (and manufacturer).

The implementation of such systems as described above, although important, must not impinge on the safe operation of the control system, even though the diagnosis systems may involve more code and processor resource than the "operational" software.

There is a tension in meeting the requirements of OBD II (and future requirements) whilst not giving rise to false indications of impending component failure.

17. Safe states, redundancy and diversity

17.1 Introduction

In contrast to open-loop systems, and in common with all digital systems, closed-loop control systems work very reliably, but tend to fail dramatically. Therefore, it is important to emphasize the design and validation of the failure management mechanisms.

Typically, there may be more software in a production system to handle the abnormal failure mode management, than is required to handle the normal operations. This is particularly true of closed-loop control, where complex theoretical algorithms can be implemented with a few mathematical programming statements, supported by tabular read-only data.

As discussed earlier, control systems can fail for any one, or a combination of reasons, such as:

- sensor and actuator hardware failure
- sensor and actuator deterioration

- processor failure
- degradation in processor performance by noise/EMC etc.
- inadequate control algorithms and software
- unexpected extreme changes in the controlled object's behaviour.

However, the nature of closed-loop control mechanisms, is such that it may not be possible in every case, to identify in real-time the root causes of failure from the symptoms measured. So the techniques used to handle failures in open-loop systems could be inadequate in such situations. For example, the detectable symptoms of failure may be signs of instability in only some circumstances.

17.2 Analysis and design

A pre-requisite for designing a failure management system is complete understanding of all failure modes and the potential consequences. Failure mode and effects analysis (FMEA) and hazards analysis (HAZAN/HAZOPS) studies must be undertaken at the appropriate stage in the development cycle. An understanding of the consequences based on a knowledge of the control algorithms will be required and a modelling activity may be necessary to achieve this.

One difficulty is the extent to which the failure management system is designed to cope with the combinations of component deterioration, environmental extremes and hazards such as driver misuse, etc.

17.3 Failure detection

See MISRA Report 1 [11] for on-board diagnostics and failure detection techniques.

17.4 Summary of recovery techniques

The following are examples of failure management techniques that can be used once failure or deterioration has been detected (no order of preference or priority assumed):

- Enter a pre-defined safe-state or limited operation.

- Disable the facility or initiate a shut-down sequence. Note: disabling a facility may not provide the "safe state".

- Reconfigure to use a redundant sensor or actuator in a multi-channel system.

- Reconfigure to use a partially redundant sensor, by inferring input data from an alternate source of information, or start-up data. (See §12 "Diagnostics" and §16.6 "Trend logging").

- Reconfigure to a completely redundant control system, using an alternative processor, sensors and actuators.
- Substitute open-loop operation.
- Disable or re-initialize adaptive control and/or data.
- Substitute an alternative/diverse control technique in a multi-channel system.
- Turn on warning lamp/display a message to the driver.
- Store fault code/message.
- Send a fault message to an associated subsystems in the vehicle via a communications link, indicating a potential for a reconfiguration of subsystem responsibilites.

A combination of several of these techniques will be relevant in most cases. However, there is a high risk of "fault masking" in microprocessor based closed-loop control systems. This occurs when the failure mode management is so effective that the driver fails to recognize the presence of any trouble and continues to use the vehicle until a further fault arises; by which time, the consequences may be much greater. The combination of techniques selected needs to be designed to indicate to the driver the nature and severity of the fault. In critical situations, a warning lamp alone may not adequately convey this information.

17.5 Commissioning and validation of failure management systems

Calibration parameters are frequently employed to optimize the performance of failure management mechanisms. Where the application is safety related, it is important to adequately resource the commissioning, parameter calibration and validation activity. In fact, it as important to adequately resource this activity as it is to resource the optimization of normal operating parameters. See §8.3 "Mapping of open-loop systems".

It will also be necessary to induce failures on prototypes or test rigs to undertake this work. Techniques used to accomplish this can include:

- open and short-circuit of connectors and wiring
- fitment of faulty and worn components
- use of artificially aged components
- addition of excessive loads on test rigs
- computer modelling of failure modes.

17.6 Summary

Not only will there be more program code system to handle failures than is required for normal operation, but experience has shown that it typically takes more human resource to develop the failure management system than it takes to develop the normal operating features.

Part III — Practical considerations

18. Modelling techniques

18.1 Introduction

An analysis and modelling phase is now a pre-requisite to the implementation of the software for a real-time system involving closed-loop control and feedback. Many organizations working in this field have, over a period of years, developed not only the computer based modelling tools necessary for this work, but also the control systems expertise that is needed to make the development process and the resulting product a success.

Control systems modelling is a well established practice in industrial sectors that have a high safety criticality, such as aerospace, nuclear and defence, and where the use of prototypes for analysis represents too high a risk. In the automotive industry, the use of prototype vehicle and powertrain systems has in the past been seen as a safe and economic option, but increasing use is now being made of computer-based modelling. These reasons are given for this:

- extreme environmental effects can be simulated more quickly and economically
- combinations of hardware failures can be represented quickly and economically
- tests are completely repeatable and results can be easily retained
- well supported modelling packages are now commercially available
- sufficient computer power for the task is now available on office based PCs and workstations.

18.2 Typical facilities provided by commercial packages

Data capture and analysis	Time and frequency domain analysis
Linear analysis	Linear and nonlinear simulation/models
Matrix manipulation	Continuous, discrete, polymomic and state space
Physical component libraries	Control algorithm libraries
Interactive control	Graphic user interfaces
Executable code generation	CASE tool interfaces
Documentation generation	Interface to publishing system

18.3 Examples of commercial modelling packages

These examples are listed in alphabetical order.

ACSL	EASY-5	SABER
ADAMS	MATLAB/SIMULINK	SPICE
ADSIM	MATRIX-X/X-MATH	TUTSIM

18.4 The advantages and disadvantages

The "pros" and "cons" of a commercial modelling package versus in-house developed models can be summarized as:

Commercial Package	Dedicated development
Available for a quick start	Needs initial investment
Generalizations can lead to over-complexity for the individual user	Often a not very user-friendly user interface
Can lack a feature required by the user	Can be very specific to needs
Benefits from a wider user community	Builds in-house expertise
Interfaces to other products	Needs specialist skills
Availability of upgrades	Needs on-going support
Portable across development workstations and PCs	Adaptable to become a "real world" training simulator
Compatible across company boundaries	Adaptable to become a real-time test facility
Training and documentation available	Training and documentation expensive to set up

18.5 Discussion of commercial packages

18.5.1 Introduction

There are quite a large number of commercial modelling packages available. Typically, until quite recently, it was necessary to run modelling packages on a mainframe, or at very least a powerful mini-computer. Today packages are available which run on a medium powered graphics workstation, or a personal computer. Commercial packages are designed to be "user-friendly", and employ some form of WIMP user interface; for successful use on a PC usually requires a fast processor incorporating a maths co-processor, or equivalent capability. The biggest constraint on performance in the PC environment is memory size, and the operating system software environment itself.

There are pros and cons for using modelling packages, which are discussed below; different commercial packages offer different capabilities and features. This section is compiled from information provided by two competitive packages which are probably the best known: MATLAB and MATRIX-X. Both are US products.

18.5.2 Basic facilities

The modelling package is built around a mathematical "engine"—a stand alone product, which has the capability to perform complex linear algebraic computations and matrix manipulations, Fourier transforms, polynomials, and differential equations. The package is based on a library of mathematics routines, which may be called and used in other compatible applications.

The mathematics "engine" also contains a data manipulation and analysis suite, allowing "real world" data to be imported and operated upon.

The second, major part of the mathematics "engine" is a set of graphics routines, which allow two- and three-dimensional representation in colour of complex data on a high-resolution graphics VDU or colour printer/plotter. Three dimensional plots may be represented as meshes or as colour-contour "map" surfaces. The GUI of one product allows user selection of manipulators such as sliders, buttons, etc., for interactive applications. Both products employ a macro-type language to allow for the user to design customized mathematical functions.

All other functions are provided by add-on products.

18.5.3 Control system modelling add-ons

The add-on of particular interest for this section is the control system modeller package. Both products offer the capability to design and analyse classical and modern control algorithms, in time and frequency domains.

Each product has software modules which provide for:

- control design and analysis
- signal analysis
- system identification
- robust control procedures
- optimization.

18.5.3.1 Control design and analysis

Classical control design tools provided are typically:

- root-locus, Bode, Nyquist, Nichols plots
- gain and phase margin calculation
- pole-zero calculation.

Modern control tools include:

- pole placement
- LQR/LQG/LQE

- controllability/observability
- solution of the Riccati equation
- calculation of eigenvalues/eigenvectors
- model reduction.

System representations for dynamic systems include state-space, transfer function, gain-pole-zero. Systems may be analysed using impulse response; step response; ramp; steady state signal simulations by frequency-response (including multivariable), root-mean-square, PSD, and Lyapunov (state covariance) methods.

The techniques listed above, which are by no means complete, cover most of the needs of modern and classical control design and analysis. Most nonlinearities can be modelled, however, accuracy of a nonlinear model needs to be carefully considered if serious nonlinearity is involved.

18.5.3.2 Signal analysis

In order to create an accurate model, it is necessary to be able to model input signals, in order to simulate plant response, manipulate output signals; this is carried out in the signal analysis add-on.

The signal analysis add-on allows modelling of various filter types, including digital and analogue types, FFTs and other transforms. Linear and nonlinear channels may be simulated, as well as a variety of signal and noise generators, and modulation and demodulation. This add-on is principally used for processing data input into the mathematics package, and can perform such functions as spectral analysis and random noise injection.

18.5.3.3 System identification

In principle, the main purpose of this add-on is to identify a system's characteristics from analysing its input and output data. From experimental data, a system's transfer function or state-space model may be computed. The model may be arrived at using a variety of techniques, including recursion, and spectral analysis.

18.5.3.4 Robust control module

This add-on is used to improve the design of systems in order to achieve guaranteed stability margins where the system contains uncertainties. Models often demonstrate a better level of performance expectation than the actual system delivers when it is realized, due either to plant uncertainties, or actuator and sensor uncertainties (or both) not being taken into account in the design model. The robust control add-on uses LQG/LTR, H_∞/H_2, or spectral factorization techniques to estimate or compensate for uncertainties. It is possible to use this add-on to identify alternative, more robust control strategies to improve system performance. μ_ analysis and synthesis is also possible.

18.5.3.5 Optimization module

The optimization module is designed to speed the process of optimizing either linear or nonlinear control systems. Cost, constraint functions, and tolerances and iteration parameters may be specified for customized optimization. Parameters which are to be used to optimize the system response may be specified, along with the constraints and cost functions, and the module used to adjust the system control parameters to achieve the desired performance.

18.5.4 Simulation/animation module

A powerful feature of modelling packages is the use of the graphical user interface to provide a simple means of creating a model of a required system, and animating it with either computed or "real world" data. This graphical approach to modelling is based on a library of standard "building block" circuit elements. These may be linear or nonlinear, and have their parameters set via a mouse driven user interface employing pull-down menus and dialogue boxes. User defined functions are also allowed for. The advantage of the graphical approach is that no programming is required, thereby, theoretically, allowing lesser-qualified staff to design and model complex control systems. As well as animating the "system", its behaviour may be monitored by a variety of software-based "virtual instruments", such as simulated voltmeters and oscilloscopes.

All types of nonlinearities may be simulated, including clipping, deadband, hysteresis, two-state ("bang-bang"), and gain table defined. There are various "tools" incorporated to allow inspection of the "system's" behaviour as the simulation progresses, including allowing changing parameter values during a simulation.

In general, animation is based on timed events, synchronous with the system clock. One of these packages now offers asynchrous (i.e. interrupt driven) animation, allowing improved modelling ability for systems such as engine management or ABS.

18.5.5 Automatic code generation

After a system is designed, modelled and optimized it is often desirable to "try it out" controlling the real plant, or using the target processor. The automatic code generator module provides this facility, by automatically generating source code based on the building blocks identified in the simulation module. Typical source code languages include C, C++, ADA, and Fortran. Code is claimed to be optimized, and to run in real time. One auto code generator has an automatic code validation facility built in.

From both products, links may be made to other software—whether routines within the product's other modules, or maths engine, or to user-written routines. The code may be manually optimized further, or adjusted, although this tends to negate the benefit of an automatic code generator. Code is automatically commented.

The code may be used with "real world" data, on a target system, or with design data, and/or from the workstation.

18.5.6 New technologies

One of the products considered offers fuzzy logic and knowledge based system design add-ons (i.e. rule-based), the other a neural network add-on.

Fuzzy logic, or other rule-based systems, are principally used to mimic the actions of an operator—which are predominantly qualitative processes, or for defining systems with imprecisely defined dynamics. There is clear applicability for a modelling and simulation package in such an arena, to aid in acquiring data for setting the rules, and assembling a system model to represent the rules.

Neural networks, on the other hand, are a learning system; good for systems containing unquantifiable nonlinearity, high complexity, and where rules would be difficult or impossible to configure. It is essential to try out a neural net system with, for example, different target number of learning cycles, different data spreads, different rates of adjustment of synapse weightings. This can be a very time-consuming process indeed, which can be facilitated quite considerably using a modelling package.

Both the above modules allow modelling and rapid prototyping of system code for rule based and self learning systems, and incorporate "tools" for graphically assessing performance, etc.

18.5.7 Miscellaneous facilities

Both products offer additional statistical and mathematical add-ons, as well as splitting the features of the add-ons listed above in a different way from each other. These miscellaneous products are beyond the scope of this document.

18.5.8 Validation

It is clear that both these products, and indeed other commercial modelling and simulation packages, are very powerful tools indeed. They offer in particular a quick and relatively simple way to get from a system specification to a prototype. There are however important issues to be considered:

- validation/verification of automatically generated code
- validation of models generated
- validation/verification of mathematical routines in the mathematical engine
- traceability of results.

It appears that these large and complex pieces of commercial software have not generally been subjected to formal V & V procedures, or, for example, certification by the FAA. Their manufacturers place faith in a large user base reporting faults (i.e. product maturity), and in using skilled programmers who "test the routines to destruction". This creates a dilemma for MISRA: can they be recommended?

It appears that the aerospace industry uses these products for both designing their flight

systems controllers, **and for validating those systems**. Both products are based on "standard" mathematics library routines, which are available in a validated or possibly certified form. Generally, the maths routines have been validated against known and proven mathematical tests as individual routines.

As a general principle, the correctness of a model can be assessed by comparing the outputs for a range of defined inputs from the model, with the outputs from the automatically generated code running in the target system for the same inputs. Given that all combinations of inputs versus outputs cannot possibly be tested for either the model or the target system, this neither validates the model nor the software in an conventionally acceptable manner. If the assumption is made that an error is detected, it is not clear how differentiation is made between algorithmic, specification, and software errors. One package, however, does claim to automatically validate autocode.

It is stated that autocode generated in ADA by one of these products has fulfilled the requirements of 85% of coding requirements for a recent spacecraft project.

18.5.9 So what conclusions and recommendations may be drawn?

- Powerful, user-friendly control systems design and modelling tools are readily available from commercial sources.
- There is clear applicability for these tools in the automotive environment.
- Validation of the systems' own routines is rather less than formal; however, their use seems to offer advantages, providing autocode generated is treated with caution. One of the suppliers of these packages does not recommend its use in the final product, only for rapid prototyping.
- It is clear that these products will continue to evolve and improve; it would be unwise of MISRA to deny their usefulness.

19. Simulation and emulation

19.1 Definitions

Simulation: a condition where a "real" system is exercised in a synthesis of its intended environment.

Emulation: a condition in which a synthesis of a system is exercised in its intended "real" environment.

19.2 Purpose

Simulation: As an example, during development of an engine management system, it is necessary to test the ECM under bench conditions, in order to adjust its behaviour to achieve an adequately safe and representative condition prior to an active vehicle test; to achieve basic arrangements to diagnostic schemes to enable diagnostics development to take place; to simulate fault conditions experienced under road test conditions in order to aid diagnosis and rectification.

Emulation: Again using an engine management system as an example, it is necessary to set up basic fuel-loop data in order to be able to assess the needs for the ECM, and to try fuelling and ignition timing adjustment during road test in order to achieve a basis from which to carry out development. For this purpose, controller emulators can be used, which permit rapid adjustment of parameters, or their effect or scaling relative to engine behaviour.

19.3 Simulation

Simulation may be implemented as a computerized model of the desired environment, or be a simple set of adjustable loads, signals, and indicators, designed to offer a non-realtime related "hook-up"; or some intermediate combination of the two. Plainly, the very simple arrangement which is fully manual in operation cannot be considered as relevant to the discussions required by MISRA objectives, albeit a popular tool with engine management developers. It must not be viewed as a tool which offers a realistic assessment of the behaviour of a system from a control-theoretical or algorithmic point of view. It may offer a convenient way of checking setting of simple parameters, or the reaction of, for example, an off-board diagnostic tool to a set of predefined conditions.

The case of using a computer model of the host environment, however, needs careful consideration.

The important question to be answered is "how accurate is the model?" Added to this is the question of how to prove how accurate the model is, and possibly how accuracy is to be determined.

It must be accepted, if a simulation is to be accurate, that it will probably require as much effort as designing the controller—maybe even more. It is therefore important to understand **why** a simulation is to undertaken; if it is merely a matter of convenience, the cost may be unjustifiable. If it is because there is no other option, then the simulation **must** receive the level and standard of care and rigour in its design as the criticality of the target system would be expected to demand.

This raises a further issue: if it is necessary to simulate the host environment because the actual environment does not (yet?) exist, how will the accuracy of the simulation be

determined? Will assumptions be made that cannot be proved in practice? Have all aspects of the host environment been properly and reliably defined in a form compatible with the format of the model?

In practice, the simulation will require to be an accurate implementation of the requirements specification of the system it is simulating. It is essential, especially where a safety-critical system is involved, or safety-critical aspects of a system are being tested by simulation, to operate a validation and verification process for the simulator which is traceable to the requirements specification, or at very least, for which the compromises adopted are known, accurately defined in quantitative terms, and documented.

Over and above implementing the requirements specification is a requirement to simulate the dynamic characteristics of the device being controlled. This is a rather more difficult task—especially if the device does not exist at the time of design of the simulation—and is particularly important when evaluating the behaviour of a feedback control system. Again, it is essential that the dynamics of the device being controlled are accurately described by the simulation, if the simulation is used to validate, for example, stability of a control scheme, with any reliability. The dynamics of the device being controlled are supplemented by the dynamics of any sensors or transducers used in the system control loop: these too must be accurately simulated.

Control systems present a particularly difficult task for simulators. It is essential for nonlinearities to be accurately simulated if the the controller is to respond to its simulated environment in a fashion representative of its behaviour in its actual host environment. If nonlinearities due, for example, to sensors, are compensated for by manual adjustment of parameters within the algorithm, significant errors can be introduced by using poor simulations of the sensors. The same is also true of the device being controlled. If nonlinearities are simulated by locally linearized quantities, the range over which the simulation is valid must be carefully observed and maintained.

It must be concluded from the above that the use of simulation for design, development, and particularly, validation purposes in complex control systems is not a trivial task, and is every bit as important as the design of the control systems itself if meaningful results are to be obtained.

19.4 Emulation

In the case of emulation, it is the dynamics of the controller which must be reproduced, if meaningful results are to be realized.

An emulator is, in most cases, basically a computer with its own operating system, which is used to mimic the operation of another computer. This immediately indicates that it will probably have a different response characteristic to its stimuli than would the computer it is attempting to emulate.

In automotive applications it is more usual to emulate RAM than the processor itself, since the most usual use of emulation is to aid the process of optimization. A good example of this is driveability of engine-managed vehicles: there is a safety implication in being able to interfere with the processor or control algorithm whilst in motion. Access to, and modification of values in RAM, however, is acceptable. This type of emulator is principally aimed at assisting the calibration engineer.

Emulators have the biggest impact on systems timing: in critical real-time applications, the emulator will either have to have considerably more speed than the processor it is emulating—and efficient operating system code—or else it will not perform adequately.

Single chip controllers also need to be considered; the separate section below should be consulted. However, it is difficult to adequately emulate more than the RAM or the program memory.

If an emulator is employed to adjust parameters of a control algorithm, there is a significant risk that it will not behave in a representative fashion, and that the results will be erroneous, requiring further modification of the actual system when it is available for service.

The use of emulators, if properly contrived, can help to identify combinations of inputs and outputs which cause system problems.

They can also avoid the possibility of software problems causing hardware failure in the target system. An emulator can help in all aspects of trying the software for a control system in conditions in which it will be expected to perform, with the possibility of feeding back information about input/output states, etc., except for fast time-critical functions.

20. Single-chip controllers

Single-chip controllers may offer analogue and digital I/O, on chip floating-point co-processor, serial data link, clock, and EEPROM in one package. This is attractive for some automotive applications, not least because of cost and size, but it carries penalties, the chief of which is the added complexity in trying to emulate the system. No longer can just the central processor be emulated.

Emulation of the whole chip is probably too expensive and complex to consider, therefore it is most usual to emulate just the on chip RAM, and/or ROM. Some single chip controllers offer connection to the data and address busses for this purpose, as well as that of connecting additional I/O, RAM, etc.

Recently it has become possible to purchase single-chip controllers of the above type, with windowing-based software implementation of a fuzzy-logic set, which operates as an emulator for development purposes, and can download software into on-chip EEPROM. It is suggested that this is aimed at automotive control, among other commercial applications. The suitability

of such an approach must be treated with extreme caution if it is envisaged to use it in a safety-related application.

21. Data acquisition

21.1 Introduction

The purpose of this section is to examine the issues in data acquisition for feedback control systems. For the purposes of the present section, "data acquisition" means the process by which a controller gathers input signals from the sensors and prepares them for processing. The term is used in other contexts within the motor industry which are outside the present scope. They include

- acquisition of experimental data for use in modelling and development phases
- acquisition of data for use in the system identification phase.

This section should be read in conjunction with §7 "Sampling and aliasing", and also with §14 "Accuracy and scaling".

It is assumed that those involved are practising control engineers with evidence of appropriate education [5, §7].

21.2 Method of acquisition

The correct hardware for the task must be chosen to ensure that the acquired data is within the specified accuracy. In particular, the analogue to digital converters (A/D) for the inputs, and also the digital to analogue converters (D/A) for the outputs, must have sufficient resolution and accuracy.

If acquisition and processing delays are too long, the delay between input and output can lead to "out of date" data being used to control the plant. The choice of the sampling rate, and the relationship to the bandwidth of the plant, need to be considered.

21.2.1 Multiplexing

Multiplexing of input signals is sometimes used in control systems, whereby **one** A/D is connected in sequence to a number of inputs. This has the advantage of reducing component costs, but the disadvantage of introducing "skew", where inputs sampled at nominally the same instant are in fact spaced out in time. If possible, a single A/D per channel or burst mode acquisition should be employed, but if multiplexing must be used the acceptable limits of skew should be defined in the specification.

A further consideration with multiplexing is the settling time of components such as A/Ds and

signal conditioning. In general, the analogue signals will have quite different values. A worst case approach must be used to ensure sufficient time is allowed for the hardware to settle with each new signal.

21.2.2 Burst mode acquisition

If skew is a significant problem, burst mode acquisition can be used. All required samples are taken at effectively the same instant in time. This is usually accomplished with a multi-channel A/D, which has a sample-and-hold circuit on each input, connected to the A/D circuit via an internal multiplexer. At the sampling point, all input signals are held by the samplers and then converted in turn.

21.2.3 Interrupts

It is not possible to perform synchronous data acquisition or handle asynchronous events without the use of interrupts, unless continuous polling is implemented or very fast (and consequently expensive) hardware is used. It may therefore be necessary to use interrupts as part of the data acquisition system. However, the use of interrupts in safety-critical software is discouraged by the wider community. The MISRA Noise, EMC and Real-Time report [10] gives guidance on this subject.

21.3 Speed of processing

The delays in data conversion and processing must be considered, as has been discussed in elsewhere in this report. Although faster hardware can often be specified to alleviate problems associated with processing delays, there are several disadvantages:

- cost
- power supply and dissipation
- EMC
- need for associated fast memory.

In high speed systems it may be necessary to consider the use of digital signal processing (DSP) as a cost-effective solution.

21.4 Conclusions

The data acquired must be accurate and up to date, so the correct specification of hardware and control algorithms must be used to ensure an appropriate yet cost-effective solution. Suggested techniques include

- multiplexing (acceptable skew must be specified)
- burst mode acquisition
- interrupts

- DSP.

22. Sensor variability

22.1 Introduction

The issues of sensor accuracy in terms of scaling and linearity have been addressed in the relevant section of this report.

The purpose of this section of the report is to examine other factors which affect sensor accuracy and reliability. Different types of sensor exhibit variability which depends on their type and construction; often similar types of sensors exhibit different variability due to differences in construction only.

Sensor variability has become an issue primarily because greater expectations of vehicle electronic systems—for example, exhaust emissions and fuel economy—has created a demand for better sensors. At the same time, improvements in technology have increased the possibilities for novel sensor designs.

22.2 Types of vehicle sensors

Variability is closely related to the technology employed in an individual sensor; the list of sensor types below is representative of the range of technologies employed, but is not necessarily complete.

- pressure/vacuum
- speed
- position
- "knock"
- temperature
- acceleration
- motion
- mass air flow
- fuel flow
- MAP.

The oxygen sensor is not relevant to this discussion, and is therefore not included in the above list.

22.2.1 Pressure and vacuum sensors

These sendors traditionally operate by stressing a diaphragm, the movement of which activates a mechanism to convert movement to voltage. The mechanism may be a potentiometer, a

strain-gauged beam, or a piezo-electric or piezo-resistive device. Some sensors use a piezo-resistive "slice" as a diaphragm, with no intervening metal.

Failure of pressure or vacuum sensors is often equatable to failure of the diaphragm. Prior to catastrophic failure, however, the diaphragm often contains cracks, which

- allow potentially damaging fluid into the electrical/electronic part of the sensor
- alter the characteristics of the sensor, sometimes quite dramatically.

Similarly, a sensor's performance may be compromised by fatiguing of the diaphragm with age.

The electrical or electronic components of a sensor may also suffer ageing effects, or effects related to material quality or workmanship. A classic case is the potentiometer based pressure sensor, commonly used for example for measuring oil pressure, which suffers degradation of performance with age due to mechanical wear of the sliding contact, resistor element, and the pivots of the sliding contact assembly. The effects of this wear include intermittency, change of resistor characteristics, and electrical noise. On electronic sensor elements the effects are more subtle. Very few strain-gauge sensors are employed on vehicles because of their high cost; however, piezo sensors are common. An unprotected piezo material diaphragm may suffer erosion of the material by some automotive fluids, with a resulting change in sensitivity, linearity, or both.

22.2.2 Speed transducers

Systems which have a requirement for speed measurement include engine management, ignition, ABS, traction control, transmission control, ride control, active control of four wheel drive, steering effort control. All these systems necessitate the measurement of the speed of rotating components.

The speed of rotating components may be measured by optical or magnetic means. Essentially both techniques involve counting teeth on a toothed wheel, in some form. Optical means are normally only employed where the environment for the sensor is inherently clean.

Optical speed sensing can be implemented in one of two ways: either an opaque toothed wheel or clear disc printed with an opaque pattern, rotating between an LED and photosensitive device; or a rotating reflective element, accessed by a light source and photosensitive device from one side only. The latter method is usually confined to test instrumentation.

To maintain integrity, it is essential that the light path between LED and photosensor is not degraded. Degradation can be due to both reduced light output from the LED with ageing, and due to the ingress of dirt into the light path. The effect of this degradation is late recognition of a light pulse, or missing pulses. This may be of special importance where directional information is required as well as speed.

Magnetic speed sensing is popular in automotive applications. It does not suffer degradation due to ingress of dirt into the magnetic path, although ingress of metallic debris may have detrimental effect. The greatest limitation on magnetic speed sensing arises out of the fundamental nature of magnetic induction: the output signal amplitude from the sensor is dependent on the rate of change of flux, and therefore, speed being measured. At low speed, pulse amplitude is low; it is usual to design pulse shaping circuitry to cope with the lowest amplitude signals from the sensor, with higher amplitude signals being clipped at a predetermined amplitude. Sensor degradation is probably restricted to weakening of the internal magnet, resulting in reduced output amplitude. This will have the effect of causing loss of pulses at low speed. Magnetic sensors are often used in conditions of severe vibration and shock, as in ABS wheel speed measurement, and need to be of high integrity construction. Vibration may cause failure of "potting" of the coil, or the weather sealing between coil assembly and shell; if this results in water, or for example, salt, ingress, the most likely result is failure of the coil, rather than degradation. The sensor's connector is likely to be the major source of reliability problems, for instance corrosion in the connector could cause reduction of signal amplitude, or intermittency (missing pulses).

Of greater concern are magnetic sensors incorporating signal conditioning circuitry. This is often a miniature thick-film circuit assembly, which will be vulnerable if weather sealing fails, and water enters the sensor.

In general, however, magnetic speed sensors' behaviour (frequency versus speed) is determined by the number of teeth on the rotating device and its speed range; this cannot, of itself, change or degrade.

22.2.3 Position sensors

Position sensing is required for throttle-angle measurement; ride height measurement; gearshift indication; automatic-clutch position; seat/mirror position; fuel and oil levels; air conditioner flap positions; ignition advance angle; crank position; steering wheel angle.

The latter two quantities are usually derived from speed sensor installations, and as such are largely dealt with above (speed sensing); reduction or degradation of the output signal of magnetic sensor may result in variability in the positional measurement. This is usually quite small, and usually may be neglected.

Most position sensing in automotive applications is by resistive potentiometer. Two types of potentiometer are most often encountered: wirewound and conductive plastic element types. Both types suffer degradation as a result of prolonged use (wear), and both may be susceptible to being affected by any fluid ingress, especially salt water. Wirewound types are rarely used for position sensing, except for low-criticality applications such as fuel or oil level indicators, where limited accuracy is unimportant. The same comments are true as for resistive-element pressure sensors above regarding degradation. Resistive element sensors are acknowledged to be of less than acceptable reliability in critical applications. Reliability is often improved by using multi-track potentiometers, multi-contact wiper assemblies, or a combination of both.

Position sensing may also be achieved using optical means. Except for ignition advance angle, this is principally used only where high accuracy is required. High accuracy applications demand the use of shaft or linear encoders, which are very expensive, and usually beyond the scope of automotive applications. Reasonable accuracy may be achieved by using relatively simple, cheap printed clear plastic devices for positional encoding. As with speed sensors, the enemy of optical devices is dirt, though there is an added potential for wear in mechanical parts allowing contact between the optical parts and the encoder, with resultant scratching of the encoder giving degraded performance.

An additional type of sensor which could be employed in automotive applications is a differential voltage transformer, available in linear (LVDT) and rotary (RVDT) versions. These sensors are generally not susceptible to wear (no moving electrical conacts), and are regarded as very stable. Typically, their cost has limited their use to test instrumentation applications, however, the need for higher reliability in automotive applications may influence their usage in the future. As in the case of potentiometers, the rotary version can cope with operating angles up to approximately 270°, whilst linear versions can offer ranges of travel up to at least 10 centimetres.

22.2.4 "Knock" sensors

"Knock" sensors are employed for critical control of ignition advance, so that an engine may be operated very close to the onset of detonation. "Knock" sensors are tuned-mass accelerometers, designed to respond to the range of frequencies inherent in detonation ("pinking") which are transmitted through the engine block. The sensor is mounted at a point on the block which is computed to be the optimum point for detection of detonation frequencies in all cylinders.

Generally, degradation or failure of the sensor will manifest itself as an increase in detonation (in systems employing "knock" sensing, there is often momentary detonation while the ignition advance is modified).

This type of sensor is used in a hostile environment, and must be of high quality construction if it is to be reliable. Probably the greatest limitations on reliability are its connector and harness, and its attachment to the engine block. If the latter becomes loose for some reason, clearly the behaviour of the sensor will be seriously affected. Connector quality is important, since on many engines, it may be doused with hot, dirty engine oil in the event of an oil leak; may be subjected to steam cleaning with detergent cleaners; may be doused with fuel if there is a leak; may be subjected to salt water spray, and may possibly encounter brake fuid in the event of a leak, or spillage during replenishment. In addition the connector is subjected to cyclic heat/cooling together with continuous variable frequency vibration.

This sensor carries a safety implication, if it fails, when fitted to a car with a basic over-advanced ignition setting which is retarded by reference to the knock sensor: it could, if the failure went undetected, result in serious piston damage, with oil entering the combustion space leading to a fire. It must be stressed, however, that a fire would only happen if very serious engine damage had already occurred—and most drivers would not fail to notice that

something was wrong before serious damage was caused.

22.2.5 Temperature sensors

The following types of temperature sensors are manufactured:

- heat-variable resistive (thermistor)
- metallic (e.g. PRT)
- thermocouple
- semiconductor
- bi-metal switch.

Most temperature sensors used on vehicles are of the thermistor type. These usually comprise a thermistor bead contained within a metal enclosure, immersed in a heat conducting paste. The enclosure usually contains a connector, though some are produced with flying leads.

The most common variability comes from ingress of fluid into the enclosure, usually after a long period of service. This may cause cavities in the heat conducting paste, and may corrode the thermistor element wires, or simply partially short-circuit it. Failure usually follows quite rapidly after fluid ingress starts.

PRTs are relatively expensive, and are not normally used on vehicles. They are also rather fragile, and not ideally suited to continuous automotive use. They are, however, a precision device if used correctly.

Thermocouples are relatively low cost, and are sometimes used in applications such as monitoring air conditioner temperatures, (e.g. air output). They are accurate, and very stable, and do not need special consideration from a variability point of view. However, their output voltage is both somewhat nonlinear with temperature, is very low level, and also requires a cold junction for values to be recorded relevant to 0° C, rather than ambient. Their low sensitivity can render them susceptible to electromagnetic interference (EMI), either directly, or in their signal conditioning circuitry.

Semiconductor sensors are used in vehicles for such applications as outside air temperature for ice warning devices; in air conditioners; and for some engine temperature measurements (eg mass air flow-see section below). Semiconductor temperature sensors are nonlinear. They are not a precision device, and may exhibit variations in performance between different individual sensors. They are relatively susceptible to EMI.

22.2.5.1 Catalyst temperature

This is a legal requirement for the Japanese market, where the type of fuel used (Indolene) can promote high catalyst temperatures. Failure of this sensor could result in a fire, if a catalysed vehicle were to be parked on tall grass.

22.2.5.2 Bimetallic sensors

Bimetallic sensors are simply switches, in which one of the contacts is made from a a bonded together pair of metals with dissimilar expansion properties. The contacts are shaped such that there is a "snap" action change of state when the temperature to qhich they are exposed exceeds a design value. This type of switch is used to switch electric radiator fans, and occasionally for applications in the air conditioner unit. They may be monitored by an ECM, but very rarely are used only as a sensor. Reliability is very good, and the contact switching temperature is normally very consistent.

22.2.6 Acceleration sensors

Acceleration may be measured for use by the airbag system, in some ABS systems, and in some active suspension/active rear steer systems.

Acceleration sensors take three forms: moving mass types, piezo-electric types, and fibre-optic types.

It is most usual to employ moving mass types in vehicle applications, except for some air bag systems, where inertia switches are used. It is also most usual for the types used to contain signal conditioning circuitry. Output is usually an analogue voltage, though digital types exist. There is little variability, since the technique used is fundamental in nature—effectively a mass suspended on a spring, operating as an LVDT. Ageing may result in increased sensitivity, as the spring fatigues, but this is unlikely to be a serious problem during the life of a motor vehicle.

Piezo-electric types are very similar, except that the seismic mass is suspended on a block of piezo-crystal or piezo-ceramic material. As the seismic mass is accelerated, the piezo-electric material is compressed or extended, and thereby generates an output voltage which is proportional to acceleration. Piezo-electric types are very stable indeed, however, are expensive, so are very rarely used in automotive applications.

There has been research into the use of fibre-optic acceleration sensors for automotive applications in recent years. The use for which they have a particular application is as a yaw rate sensor for vehicles employing full-active suspension and rear steer. Such vehicles have been largely the product of research and racing programmes until recently, but are now nearing production in more expensive vehicles.

The theory of the fibre optic yaw sensor is that, if light is transmitted down the fibre optic cable, and the cable is accelerated, the light will arrive at the far end of the cable at a different time than would have been the case if it had not been accelerated. By comparing a "static" fibre optic cable signal with one subject to acceleration, a signal can be generated, using the Doppler radar principle, proportional to the acceleration applied. The "static" signal is in practice arrived at by virtue of the orientation of the reference cable.

Stability and accuracy of these systems is believed to be extremely good, and there is little likelihood of variability.

22.2.7 Motion sensors

Motion sensors are velocity sensitive. The only application at the moment is pitot tube speed sensors on racing cars. They need not concern this report unduly. They are a form of pressure sensor.

22.2.8 Mass air flow sensors

It is necessary for engine management systems to calculate the mass of air/fuel mixture being ingested by the engine in order to arrive at the correct fuelling. Two types of mass air flow sensors have been used for this purpose: moving vane, and solid state "hot-wire" anemometric. Both types in fact measure air velocity, and mass has to be calculated by the ECM based on intake air temperature and venturi area, which permits a measure of volume, the mass being calculated by also calculating density from the temperature measurement.

The moving vane type sensor usually operates a potentiometer, which is clearly, in itself, stable. However, the swivels of the vane may corrode, and ultimately cause sticking, if maintenance is inadequate. The potentiometer wiper(s) and track are in constant use, and are therefore subject to the effects of wear. The anemometric type is vulnerable to contamination from dust and oil vapour, the latter drawn back into the manifold by induction system pressure pulses.

22.2.9 Fuel flow

Some vehicles still use turbine fuel flow metering for trip computing, however, most now measure injector pulse train frequency and injector "on" time.

Turbine fuel flow meters are not especially accurate over a wide range of flows, therefore their accuracy will be very dependent on the type of driving undertaken. Since their use in trip computers is as customer convenience devices, they need not concern this report unduly.

22.2.10 MAP Sensors

In some engine management systems, mass air flow is computed from throttle opening and manifold depression, the latter measured by means of a manifold absolute pressure sensor (MAP). This sensor is an active device, usually employing a piezo-electric diaphragm, and a thick-film signal conditioning circuit. It is normally connected to the manifold via a small bore tube. Variability can arise because of oil film and possibly dust entering the tube, and congealing, so restricting the air flow. This type of sensor is usually sensitive to the resonance of air movement in the tube, and any obstruction will inevitably change this resonance. If oil becomes trapped in the tube, it alone may be sufficient to modify the behaviour of the sensor, since, again, it will both modify the tube resonant frequency, and damp the movement of air in the tube. It is essential that this type of sensor is carefully sited to minimize the possibility of oil being drawn into the tube. It should also be noted that this type of sensor is rather sensitive to EMC.

22.3 Conclusions

The range of available vehicle sensor types has been described above, but, although fairly comprehensive, may not be complete, especially as technology moves forward. In general, sensors are remarkably stable and consistent, and most problems stem from poor quality construction, or perhaps unsuitability of an individual type to the automotive environment.

Variability of sensors is not generally a problem in the automotive environment, however, as requirements become more stringent with increasing legislation, must not be ignored.

23. Support

23.1 Introduction

The purpose of this section is to examine the issues in product support of feedback control systems. Each subsection covers a different aspect of what is a very diverse subject.

It is assumed that those involved are practising control engineers with evidence of appropriate education [5, §7].

23.2 "Chipping"

23.2.1 What is "chipping"?

"Chipping" refers to the practice of reprogramming electronic control systems to alter the performance from the manufacturer's original specification. As "chipping" is unauthorized modification of computer software, it is another manifestation of hacking. Although "chipping" can be quite sophisticated, so can other forms of hacking. The practice almost universally pertains to electronic engine management systems, where it is popularly known as "hot chipping", and therefore the remainder of the discussion will be with specific reference to such systems.

Some vehicle manufacturers do have authorized users of modified engine management systems, typically for rally or off-road uses. Similarly, different power variants of the same basic engine, for example in different models, may only involve different engine calibrations. The legitimate use of modified systems needs to be recognized, but the issue under consideration in this report is **unauthorized** modification.

In the case of engine management systems, "hot chips" are supplied which claim to increase the power output by a certain amount, improve driveability, remove flat spots and so on. They may be associated with mechanical modifications.

Historically, owners of cars have boosted performance by fitting improved components such

as carburettors, cylinder heads and exhausts. With electronic management of fuel and ignition some of these opportunities have been removed. It is to be expected that performance oriented motorists will look for other ways to boost output. In this report, the practice of "chipping" will be examined critically.

"Chipping" is usually achieved by supplying a replacement ROM or EPROM for the electronic control module (ECM). This may be fitted by the owner of the vehicle or by the supplier. On occasions, a complete exchange module may be required.

The supplier of a "chip" will usually modify the control system by a process of trial and error, where values stored in the calibration data will be changed and the effects observed. The program code itself may be modified also, one example being a supplier of engine "chips" who state that they "remove the over-run cutoff" on every car. There have been instances of more sophisticated "chipping" where fault codes have been masked and checksums changed.

23.2.2 Is it desirable?

The historical reasons for "chipping" have already been alluded to. These practices predate exhaust emissions legislation or relate to off-road use and have therefore been legal. They have also tended to be the work of professional companies.

It has been argued that for the "average" car engine the ECM is a compromise for the worst case production engine. Tuning to obtain the best performance from an individual engine is the excuse for some "chipping". However, the majority of "chips" available are "off the shelf" for a particular model and not tuned for a specific engine. Whilst appealing to performance oriented motorists, there are a number of reasons why "chipping" is not desirable. The customer is usually sold a "chip" solely on the basis of its perceived advantages and as a result is uninformed of the disadvantages. These disadvantages concern the vehicle manufacturer, the end user, insurance companies and society as a whole.

23.2.2.1 Software integrity

The integrity of the software will not be preserved. Both the program and the data in an electronic control system will have been extensively validated. If either or both is modified by another party, re-validation is required to guarantee the continued integrity of the software. It is simply not possible that a "chipper" can re-validate the system after modification or follow accepted failure management procedures. Therefore software integrity will be compromised by these practices. Note that software integrity includes the ability of the software to reject noise or electromagnetic interference and to handle failures, as well as the correctness of the code itself. It has to be stressed that modifying map data alone can compromise the integrity of the system, as new values stored may be outside the bounds of the original validation. For example, one supplier of "chips" offers high performance versions which increase the maximum engine speed by 500 rpm.

23.2.2.2 Legislative requirements

Modification of the control strategy can lead to the system no longer meeting legislative requirements, notably emissions. This could occur from modification of the calibration data, or the code, or both. "Chips" which give increased acceleration will almost certainly give increased emissions, too. Again, the "chipper" cannot possibly re-certify the modified system to ensure continued compliance with statutory emissions requirements, given that the majority of vehicle manufacturer development time on engine calibrations is taken with meeting such legislation. Some "chips" are claimed to not increase emissions, but it is likely that only idle CO content is checked. "Chipping" rarely modifies the characteristics at idle.

In extreme cases, "hot chips" require replacement of the catalytic converter with a "performance" version or is removed. Given the relatively cheap cost of the parts, it is believed that the phrase "performance catalytic converter" is a euphemism for a through pipe. A vehicle so modified cannot possibly meet emissions requirements.

Some companies advertise "free flow" air filter kits to complement the "chip". It is possible that such modifications could lead to violation of noise laws (for instance in Switzerland).

23.2.2.3 Durability

Another aspect is that modifications can lead to increased wear and hence reduced lifetime. Removing the over-run cutoff may mean unburnt fuel enters the catalytic converter, damaging it. Often increased engine performance will mean a replacement engine earlier than may be expected. There is a trade-off between performance on the one hand, and fuel economy, emissions and drivetrain durability (engine **and** other components such as the gearbox) on the other. This trade-off is often not appreciated by the end user of such modifications. It is of particular concern regarding warranties, which makes the avoidance and detection of "chipping" all the more important (see below).

Associated mechanical modifications can also have durability implications. The availability of "free flow" air filters was referred to above. However, the ability of the filter to perform its primary function of removing foreign bodies is likely to be impaired. Dirt would be allowed to enter the engine, resulting in damage.

23.2.2.4 Safety

If the engine power is uprated, other components on the vehicle need to be uprated too, such as the steering, suspension, brakes and tyres. Simply replacing an IC in an ECM does not uprate such parts and could lead to a dangerous vehicle.

Reprogramming the fuelling can itself have dangerous consequences. An example is that detonation may occur at higher engine speeds. This can lead to piston damage and ultimately an engine fire. The occupant of a vehicle travelling at speed which caught fire in this way would have very little time to escape. Similarly, some cheaper air filter elements can catch fire if backfiring occurs.

23.2.2.5 Subsequent owners

A "chipped" vehicle may be sold to a subsequent owner who could be completely unaware of the modifications. Alternatively, a complete engine management system may be replaced with one giving higher performance, the vehicle run in that state for some time and the original unit replaced just prior to the vehicle being sold. The unsuspecting new owner may very quickly discover they have a damaged drivetrain, which raises questions of liability (see below).

A subsequent owner may also face insurance implications (see below).

23.2.2.6 Liability

In UK law there is the principle of *novus actus interveniens*, that is, a new act intervening. It would appear that this principle would absolve vehicle manufacturers from responsibility for damage caused by "chipping", although this may depend on what steps they had taken to warn owners against the practice. As this principle may be extended or reduced by another principle or statute, it is important that vehicle manufacturers consult their own legal advisers on what steps they need to take to ensure they are not liable for the actions of "chippers".

In contrast, the liability risk in the USA to manufacturers is greater. The situation in Europe as a whole is less clear.

23.2.2.7 Insurance

Vehicle insurance is normally rendered invalid if a vehicle is modified and the insurance company are not notified. An uninformed customer may believe they are entitled to alter the software on a vehicle but not the manifolds. A subsequent owner may be unaware the vehicle has been "chipped" until it is involved in an accident and inspected by insurance assessors. It is believed the insurance company would have to prove the owner knowingly withheld information about modifications to withdraw cover, but the clear undesirability of such a situation makes avoiding and detecting "chipping" all the more important.

23.2.2.8 Warning lamps and messages

Certain electronic systems communicate information to the driver via warning lamps and/or messages. An example is the OBD II requirement for a lamp labelled "service engine soon" or similar to be illuminated in the event of a failure in an emissions control component. Modification of a system may mean that a warning lamp is permanently illuminated, for example if the catalytic converter and its associated oxygen sensor have been removed. Cases are known where the "check engine" lamp has been masked with tape!

"Chipping" may be used in the future to bypass warning lamps. Some drivers want the ABS feature on their vehicles disabled. Simply disconnecting parts of the system, or even removing the ECM altogether, would leave the warning light permanently illuminated. This condition would cause the vehicle to fail the statutory (MoT) test, therefore the owner might

want to "chip" the ABS to disable its function but still operate the warning light as though nothing were amiss.

23.2.2.9 Reconditioned modules

There is at least one supplier of "reconditioned" ECMs, who claim to have developed the techniques to enable ECMs to be analysed, tested and remanufactured. They claim that "all products are remanufactured to the highest standards, are individually tested and uprated to the latest specification where possible." It is not apparent what the standards used in the remanufacture are, nor how they relate to OEM standards.

Although they market uprated ECMs separately, it is not clear whether a reconditioned ECM is uprated or not. Uprated ECMs are available with "non-standard fuel and ignition curves and other modifications designed to improve driveability, performance and to overcome some of the problems associated with certain vehicles." The list of vehicle problems which it is claimed that the uprated units solve include

- low speed "flat spots"
- stalling
- mid-range hesitation
- gear change "lurch".

There are a number of other suppliers of "chips" which are similarly claimed to cure known problems. However it is not immediately apparent whether solving these "problems" leads to other problems later such as durability. These "chips" will suffer from the same problems as any other "chip", including validation and insurance.

The practice of remanufacturing is likely to lead to confusion between straight service replacement units and uprated units. Furthermore, the customer would not know whether the service replacement unit supplied was a "chipped" version or not, with all the associated implications.

23.2.3 Tamperproofing and detection techniques

The designer of an electronic control system should take steps to reduce the incidence of "chipping". Some of these measures have already been described elsewhere in this report. Tamperproofing methods include:

- sealing enclosure screws
- soldering or gluing in of memory ICs
- potting or encapsulation of the circuit boards
- control of design information (even within an organization)
- password access to calibration data modifiable using service diagnostic tools
- strict controls on the supply of such tools to repairers and their use
- use of custom hardware: access to the program and data not straightforward
- checksums stored in on-processor ROM or an equivalent that is difficult to

modify

- encryption of data: again, the key needs to be difficult to access or modify and the hardware must be designed with encryption in mind from the outset to avoid significant cost penalty
- distribution of data in small quantities around the physical memory map of the processor, making identification harder
- ensuring that messages on data networks such as CAN cannot be misappropriated to modify functions
- encryption of communications data between modules.

"Chipping" can be detected by physical means, where clear evidence of tampering is displayed after unauthorized access to the control unit. It may also be detected by the system itself, for instance using checksums. An issue which needs to be considered is whether such detection should prevent the engine from being started, or to signal the tampering to an approved party at an appropriate time such as servicing or statutory testing. Detection methods include:

- wired and sealed connectors and screws
- labels sealing module case and mountings that are destroyed if removed or broken
- checksums on program and data (preferably separate)
- identifiers that need to be exchanged between modules and matched before control strategies can be entered
- time since last module power-up stored in "keep-alive" memory.

Legislation is being considered to prevent "chipping". For example, OBD II legislation in the US requires that steps are taken to ensure that tampering is either inhibited, or that clear evidence of tampering is left behind after unauthorized access to engine management systems. It is understood that legislation in the US makes it illegal to supply to the public anything that can assist in making modifications to engine management systems. This can include documentation, instructions and hardware.

In the UK hacking is covered under the "Computer Misuse Act". It appears that a case of "chipping" brought under this Act would fail on the issue of intent, as the intent to modify the program is with the owner of the vehicle. There is possibly a case on copyright or patent issues from unauthorized reprogramming.

23.2.4 Future issues

One of the problems which has been highlighted with the increasing electrical complexity of vehicles is roadside repair. Motoring organizations are finding it harder to deal with problems, as a vehicle will often have to be towed to a main dealer for diagnosis using specialized equipment. It has been suggested that ultimately problems may be cured in the field by methods that could involve reloading software into ECMs (akin to end-of-line programming). The implications of this with respect to the current practice of "chipping" need to be considered carefully. A particular concern is that "chipping" and future

developments may be used to defeat security systems built into the engine ECM. Similarly, generic replacement ECMs could be supplied to dealers and loaded with the appropriate software at the point of installation. Again, the makers of equipment to facilitate this must ensure adequate steps are taken to prevent misuse.

At present "chipping" appears to be restricted to engines. However, it is conceivable that "chips" may be produced for other electronically controlled systems in the future, such as ABS, traction control, automatic gearboxes and suspension. With the potential for increased danger due to tampering with such systems, and their future interaction over data networks such as CAN, the practice of "chipping" must be discouraged before it becomes more widespread.

Some manufacturers are considering centralized distribution of software. This may include dealers updating software in the field at major service intervals. These developments mean that sabotage and blackmail become real possibilities, with the obvious comparison to "viruses" on personal computers. Manufacturers need to have procedures for preventing such occurrences before embarking on any widespread programme of field-updatable software.

23.2.5 Summary of "chipping"

The foregoing discussion has shown that "chipping" is an undesirable and potentially dangerous practice which must be actively discouraged through design and ultimately by legislation. Although marketed as "hot chips" or "chip upgrades", the terminology used really serves to disguise the fact that this practice is nothing short of hacking. The practice is dangerous, has undesirable side effects such as emissions and reduced durability, and can be passed unseen to a subsequent owner of the vehicle along with the associated problems.

23.2.6 Recommendations

- Manufacturers should take steps to avoid and/or detect "chipping".
- Manufacturers should advise their franchised dealers to look for "chipping" on second-hand vehicles they sell, and maybe even on all vehicles at major service intervals.
- Manufacturers should seek legal advice on the steps necessary to avoid liability in cases of "chipping".
- Disclaimers are likely to be required in owner handbooks as a result
- Motoring organizations who provide vehicle inspection services should introduce checks, particularly on vehicles thought to be prone to the practice
- Future developments to the manner of embedded software distribution should only be undertaken after mechanisms are in place to prevent sabotage and other unauthorized modifications.
- *Caveat emptor*—let the buyer beware!

23.3 Documentation

There are two aspects to documentation: internal and external. There are two basic principles which apply to all documentation, however:

- it must be understandable by the target audience (who must be correctly defined)
- it must be in accordance with any relevant standards.

23.3.1 Internal documentation

Correct records must be kept of all stages in the product lifecycle. There are many aspects to documentation of a product, of which software is but one. Documentation will need to include

- design rationale
- flowcharts
- source listings and comments
- verification details
- validation results.

Software documentation must comply with accepted standards and be understandable by anyone, not just the originating analyst or programmer. The topic of software documentation is covered in more detail in other MISRA tasks.

23.3.2 External documentation

External documentation refers to material to be supplied with the product pertaining to its use and also to its maintenance.

23.3.2.1 User documentation

User documentation is aimed at the end-user of a product, typically an owner's handbook for a car. It should state the correct means of operation, and any practices that should be avoided. The location and meaning of any warning lamps or messages needs to be given, along with the circumstances under which the vehicle should be referred to the dealer. Disclaimers will also need to be provided to deal with implications of unauthorized modifications.

23.3.2.2 Maintenance documentation

This will normally take the form of repair handbooks supplied to franchised dealers and other outlets. It will include details such as

- diagnostic procedures
- meaning of fault codes

- repair procedures. For a system with embedded software this will often mean a module exchange unless the fault lies in a peripheral component such as a sensor.

There is a strong possibility that sensitive information will need to be made available, for example how to reprogramme security features if the owner has misplaced an electronic key. Tight controls will be required on the dissemination of such material.

23.4 Education and training

23.4.1 Education and experience

The staff employed on a task must have the credentials to permit them to execute their assignments successfully. "Education and experience" encompass:

- theoretical knowledge
- academic achievement
- practical experience.

Each of these is important and staff must display a proven "track record" in each. With the increasing complexity of electronic systems on vehicles, staff employed in their maintenance must be adequately qualified.

23.4.2 Training

Education must be viewed as an ongoing process, to take account of continuing developments in the field. Staff must be aware of developments which could affect their work, such as

- current best practice
- best available technologies
- legislation
- regulations
- guidelines
- standards
- codes of practice.

Staff should be encouraged to develop their knowledge in these areas. Again, this is especially relevant where electronic control systems are concerned. Manufacturers will need to arrange training courses for dealers as new systems or developments are introduced. Training will need to give clear explanations of system attributes and uses.

Training must be appropriate to the audience. Different approaches will be required for different responsibilities within an organization, such as

- senior management

- service management
- mechanics.

23.5 Change control

Most products are subject to an ongoing process of development and refinement. It is important that proper procedures for handling change are in place. There has been considerable adverse media publicity given to cases in other sectors where unauthorized modifications to software occurred. All changes must be

- correctly authorized
- traceable to their originator (and implementer if different)
- fully documented, including
- details of the changes
- justification
- revalidation.

The system must be in line with recognized quality management systems, such as BS 5750. It must be free from loopholes or other means by which the mechanisms could be bypassed. It should allow for use by subcontractors where necessary, but with the same tight controls.

The system needs to encompass the correct procedures for updating units in the field, particularly in view of proposed developments on the centralized distribution of software. Manufacturers will need to consider if software should always be updated to the latest version at major services, bearing in mind hardware changes that may have occurred in the meantime.

A related issue concerns reporting of in-service faults. At the moment procedures for reporting would appear to be limited to warranty claims. It may be appropriate to consider extending this to report any fault on an electronic control system that a dealer encounters. An action of this kind may prevent outside bodies imposing a regime on motor manufacturers that is more appropriate for other sectors.

Procedures for reporting need to have links to the design, development and validation teams. This may well include subcontractors.

23.6 The aftermarket

The practice of "chipping" is one aspect of a wider issue: the aftermarket. An "aftermarket accessory" is a piece of equipment fitted to a motor vehicle which does not form part of the manufacturer’s original specification, that is to say, is not a standard fitment or available as a factory-fit option. Certain accessories are available as so-called "dealer fit" options, but are supplied by, if not approved by, the vehicle manufacturer. For the scope of the present discussion, an aftermarket accessory is assumed to mean a piece of equipment supplied by a third party and fitted to a vehicle after it has been sold. However, this equipment may be

supplied by a dealer.

23.6.1 Liability in the aftermarket

The principal issue regarding the aftermarket is that of liability. The question is, "If a vehicle is modified and damage, injury or death results, who is responsible?". The principle in law of *novus actus interveniens* has already been referred to in the context of "chipping". The UK EMC regulations [14] already place responsibility on modifiers. They state that manufacturers of apparatus must comply with the EMC regulations, but include in their definition of manufacture any modification that substantially modifies the EMC characteristics of a piece of apparatus.

23.6.2 Validation of aftermarket equipment

If aftermarket equipment is fitted to a vehicle is must be validated to levels which are **at least** as high as those of the original manufacturer for equivalent original systems. This is of particular relevance to systems with safety implications. An example is a security system incorporating an immobilizer. This system has safety implications as it can stop the engine. Therefore it must be validated to at least the same level as the engine management system on the vehicle. Although this applies to all aspects of system reliability (climate, EMC, power supply, FMEA) it is particularly important in the case of the software development lifecycle used.

Ideally, vehicle manufacturers should have a comprehensive list of approved equipment and suppliers. However, it is recognized that there will always be a market for third-party accessories. Ultimately, products may need to be validated against a particular application, which is likely to involve certification by independent test houses.

23.6.3 Servicing

Servicing of vehicles by an independent retailer is often popular, particularly with older models where franchised dealer costs are viewed as prohibitive. As vehicles increase in the number and complexity of electronically controlled systems, it is important that service outlets are able to handle them correctly. The availability of diagnostic equipment and data will need to be considered, but the possibility of such products being used for nefarious purposes has to be avoided.

23.7 Conclusions

There are a wide range of issues affecting the support of embedded software in control systems. Major recommendations from this section include

- "chipping" should be discouraged (see separate recommendations subsection above)
- documentation should be understandable and in line with recognized standards

- a programme of continued education is required to ensure that those working with electronic systems, particularly in dealerships, are kept informed of developments
- change control needs to be managed carefully
- questions of liability in the aftermarket need to be addressed.

24. References

[1] J.R. Leigh, *Applied digital control: theory, design and implementation*, 2nd edition, Prentice Hall, 1992.

[2] G.F. Franklin, J.D. Powell and A. Emami-Naeini, *Feedback control of dynamic systems*, 2nd edition, Addison-Wesley, 1991.

[3] G.F. Franklin, J.D. Powell and M.L. Workman, *Digital control of dynamic systems*, 2nd edition, Addison-Wesley, 1990.

[4] J.R. Leigh, *Essentials of nonlinear control theory*, Peter Peregrinus Ltd., 1983.

[5] IEC 65A (Secretariat) 122, *Software for computers in the application of industrial safety-related systems*, IEC 65A WG 9, version 1.0, 26 September 1991.

[6] J.R. Leigh, *Control theory: a guided tour*, IEE Control Series Number 45, 1992.

[7] P. Katz, *Digital control using microprocessors*, Prentice Hall, 1981.

[8] G.J. Thaler and R.G. Brown, *Analysis and design of feedback control systems*, 2nd edition, McGraw-Hill, 1960.

[9] *Subcontracting of automotive software*, MISRA Report 7, 1994.

[10] *Noise, EMC and real-time*, MISRA Report 3, 1994.

[11] *Diagnostics and integrated vehicle systems*, MISRA Report 1, 1994.

[12] BS 6719, *Guide to specifying user requirements for a comnputer-based system*, 1986.

[13] ANSI/IEEE Standard 830-1984, *Guide for software requirements specifications*.

[14] United Kingdom Statutory Instrument 1992 no. 2372: *The electromagnetic compatibility regulations*.

25. Bibliography

The following additional references may also be of interest.

K. Warwick and M.T. Tham (eds.), *Failsafe control systems: Applications and emergency management*, Chapman and Hall, 1991.

B.C. Kuo, *Automatic control systems*, 6th edition, Prentice Hall, 1991.

J.R. Leigh, *Applied control theory*, IEE Control Series Number 18, 2nd edition, Peter Peregrinus Ltd, 1988.

I. Sommerville, *Software engineering*, 4th edition, Addison-Wesley, 1992.